NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

SOLUTIONS FOR RELIABLE MULTICASTING

by

David G. Petitt
September, 1996

Thesis Advisor: Rex A. Buddenberg
Associate Advisor: Suresh Sridhar

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 1996	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE SOLUTIONS FOR RELIABLE MULTICASTING		5. FUNDING NUMBERS	
6. AUTHOR(S) Petitt, David G.			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		12b. DISTRIBUTION CODE	

13. ABSTRACT *(maximum 200 words)*

Many of the applications that will be hosted on the Marine Corps' Tactical Data Network (TDN) require data to be delivered reliably from one sender to many receivers. Reliable multicast protocols are better suited for this one-to-many communication than conventional transport layer unicast protocols. These multicast protocols will have to contend with the limited bandwidth and high bit error rates of wide area links in the tactical internet. They must also adapt to the internet's changing topology, and be robust enough to survive its inevitable disruptions.

This thesis evaluates several reliable transport layer multicast protocols for their ability to deliver data reliably over a tactical internet. Because multicast routing protocols build delivery trees for these protocols, they are also evaluated.

The bandwidth saved by multicast protocols make them particularly valuable in the tactical internet. However, at both the network and transport layers, no single protocol satisfies all the requirements of the internet. Which protocols are selected for TDN depends on how the decision maker weights the requirements of the tactical internet. The types of tactical data systems will also influence the choice of a multicast routing protocol. Similarly, the reliable multicast protocols which are selected must meet the demands of the application for which they were designed while still operating within the constraints imposed by the tactical internet.

14. SUBJECT TERMS Reliable Multicasting, Reliable Multicast Protocols, Multicast Routing Protocols, Tactical Data Network, IGMP, DVMRP, HDVMRP, MOSPF, CBT, PIM-SM, PIM-DM, HPIM, TMTP, SRM, RMP, RMTP, RAMP, MTP-2, XTP		15. NUMBER OF PAGES 140	
		16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

i

SOLUTIONS FOR RELIABLE MULTICASTING

David G. Petitt
Major, United States Marine Corps
B.S., United States Naval Academy, 1984

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 1996

ABSTRACT

Many of the applications that will be hosted on the Marine Corps' Tactical Data Network (TDN) require data to be delivered reliably from one sender to many receivers. Reliable multicast protocols are better suited for this one-to-many communication than conventional transport layer unicast protocols. These multicast protocols will have to contend with the limited bandwidth and high bit error rates of wide area links in the tactical internet. They must also adapt to the internet's changing topology, and be robust enough to survive its inevitable disruptions.

This thesis evaluates several reliable transport layer multicast protocols for their ability to deliver data reliably over a tactical internet. Because multicast routing protocols build delivery trees for these protocols, they are also evaluated.

The bandwidth saved by multicast protocols make them particularly valuable in the tactical internet. However, at both the network and transport layers, no single protocol satisfies all the requirements of the internet. Which protocols are selected for TDN depends on how the decision maker weights the requirements of the tactical internet. The types of tactical data systems will also influence the choice of a multicast routing protocol. Similarly, the reliable multicast protocols which are selected must meet the demands of the application for which they were designed while still operating within the constraints imposed by the tactical internet.

TABLE OF CONTENTS

I. INTRODUCTION

A.   INTRODUCTION

Conventional transport layer unicast protocols are not well suited for reliably delivering data from a sender to more than one receiver, yet many of the applications that will be hosted on the tactical internet require this sort of communications model. The reliable multicast protocols developed to fill this void were designed for commercial networks which generally do not suffer the limitations imposed by a tactical environment. Nevertheless, the Marine Corps should adopt a commercial multicasting solution to "…leverage the commercial marketplace's investments in information technology" and ensure interoperability with commercial systems. (MCCDC, 1995) This thesis evaluates several of these reliable transport layer protocols for their ability to deliver data reliably over a tactical internet.

This chapter is designed to frame the problem of reliable multicasting in a tactical environment, and explain the approach taken to understand it. To show why multicasting is preferable over other approaches for communicating among many participants, this chapter begins with a simple explanation of multicasting. For data to be multicast reliably over an internet, several different layers of the protocol stack are involved. How each layer of the stack contributes to multicasting is explained next. The final section discusses several aspects of the tactical internet.

B.   MULTICASTING

1.   Description

"Multicasting provides an efficient way of disseminating data from a sender to a group of receivers." (Lin, 1996) Data destined for the receivers in a multicast group is sent to a single multicast address instead of being addressed separately to each receiver. Only a single copy of the data is sent by the source; the appropriate number of copies is made by each router on the path from a source to the receivers in a multicast group (Figure 1.1).
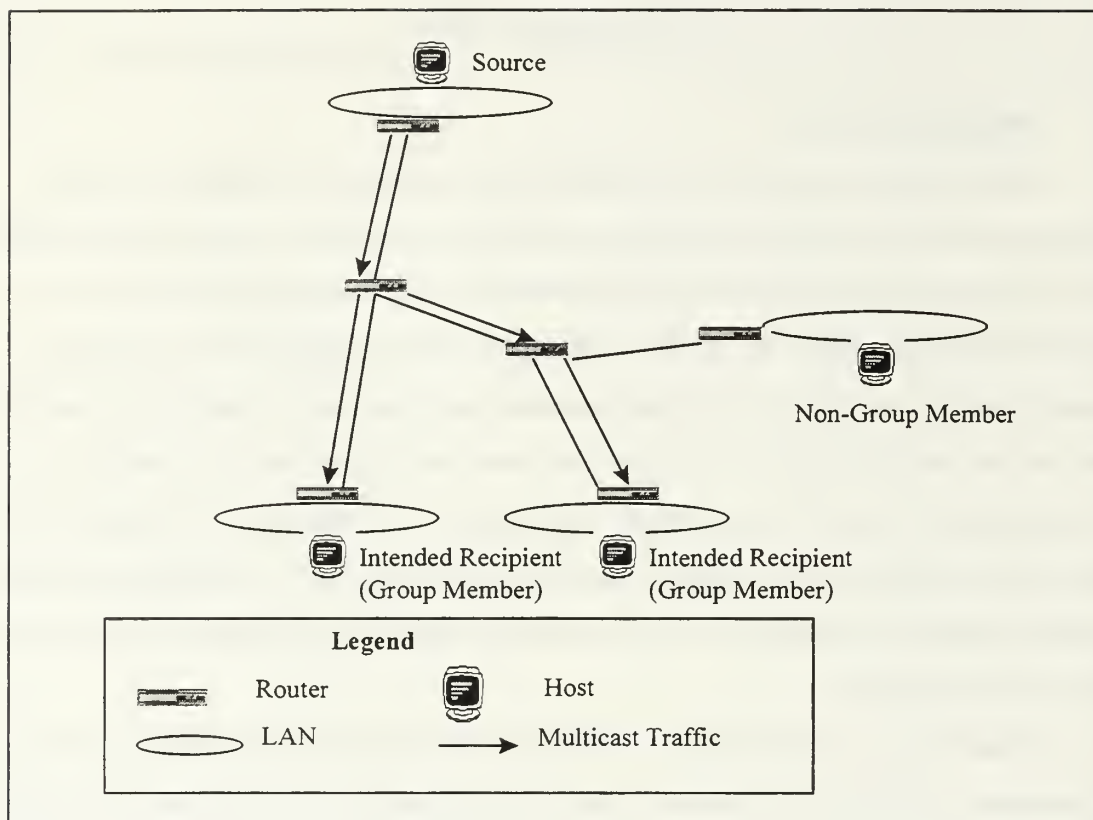
Figure 1.1. Example of Multicasting.

There are many tactical applications which could benefit from the efficiency of multicasting. Commanders at different echelons may participate in a group planning session made possible by many-to-many multicasting, where each participant acts as both a sender and a receiver. The product of this planning session may be distributed to subordinate commanders using a multicast protocol. Concurrent updates of geographically separated logistics databases may be multicast after a warehouse inventory is completed. Multicast protocols may also be used to distribute intelligence estimates or satellite imagery to intelligence consumers.

2. Alternatives to Multicasting

Multicasting is not the only way for a sender to distribute data to a group of receivers. Data may be broadcast to every receiver, or the sender may make a separate copy of the data for each intended receiver.

Broadcast packets are forwarded on all interfaces of a router except the incoming one. Broadcasting over a tactical internet is not practical since data forwarded over links not leading to intended receivers wastes bandwidth (Figure 1.2).

2
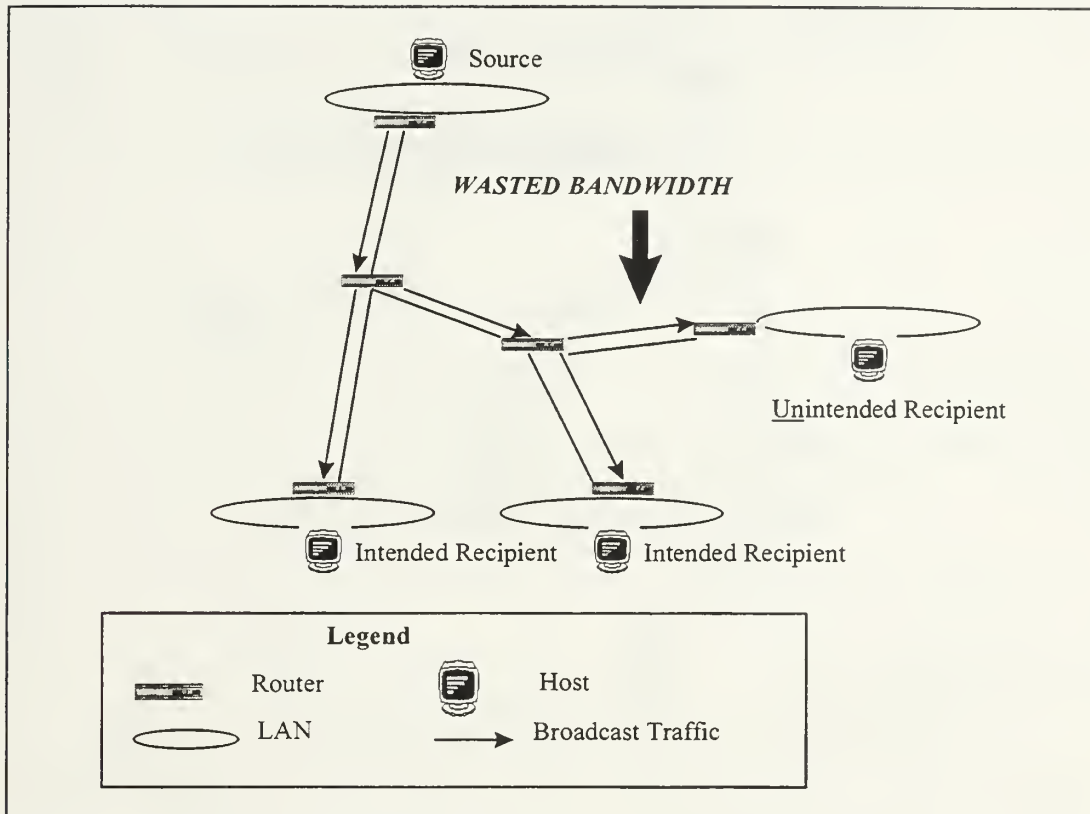
Figure 1.2. Example of Broadcasting.

Bandwidth is also wasted when separate copies of the data are made for each receiver because the same data will traverse at least one link as many times as there are receivers (Figure 1.3). Routers between the source and the intended receivers may also have to process packets for the same data more than once. Finally, "if connection-oriented service is desired, a single one-to-many connection will most likely be faster and less costly to set up than multiple one-to-one connections." (Dempsey, 1990)
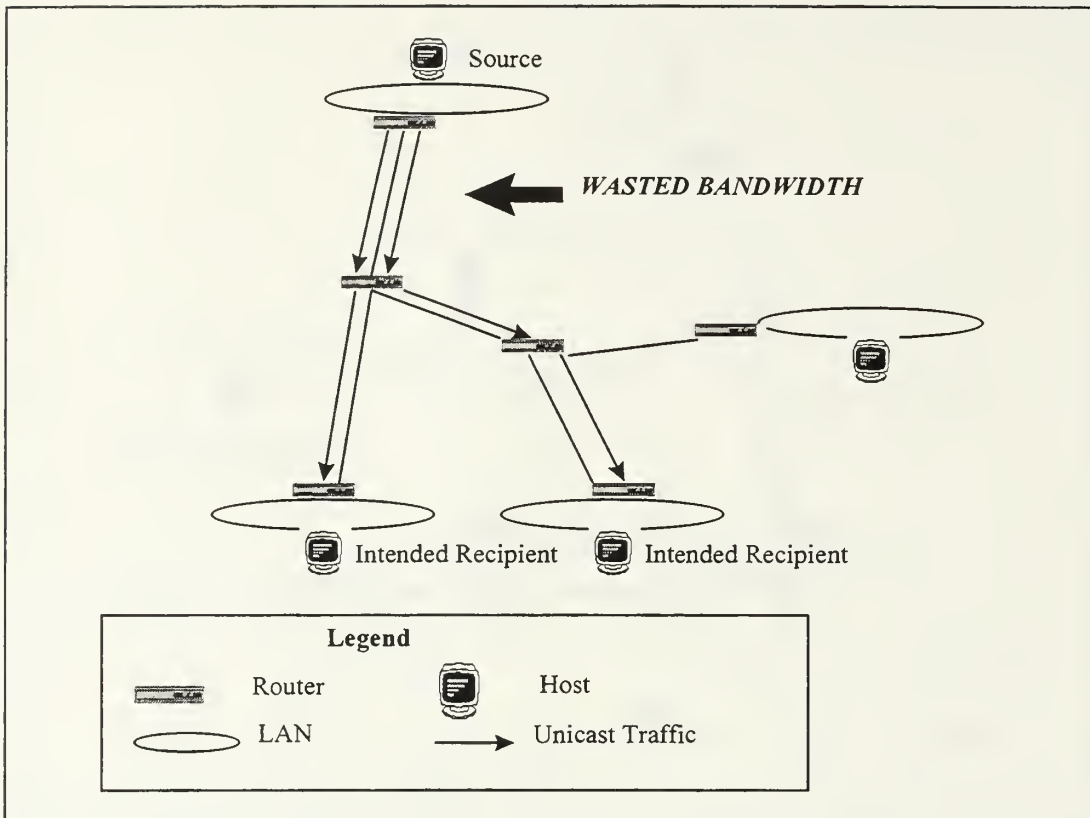
Figure 1.3. Example of Multiple Unicast Connections.

C. THE PROTOCOL STACK

1. Introduction

"The provision of a general purpose reliable multicast facility involves functionality at several layers of the ISO stack." (Dempsey, 1990) The functions related to multicasting at the involved layers of the protocol stack (Figure 1.4) are introduced in the sections which follow.

Protocol Stack	Functions Performed	What Performs the Functions
Application	→ Ordering; Managing the Multicast Group	→ Applications, Other Software
⋮		
Transport	→ Guaranteeing Reliability	→ Reliable Multicast Protocols
Network	→ Creating Delivery Trees	→ IGMP, Multicast Routing Protocols
Data Link	→ Packet Filtering	→ NIC's
Physical		
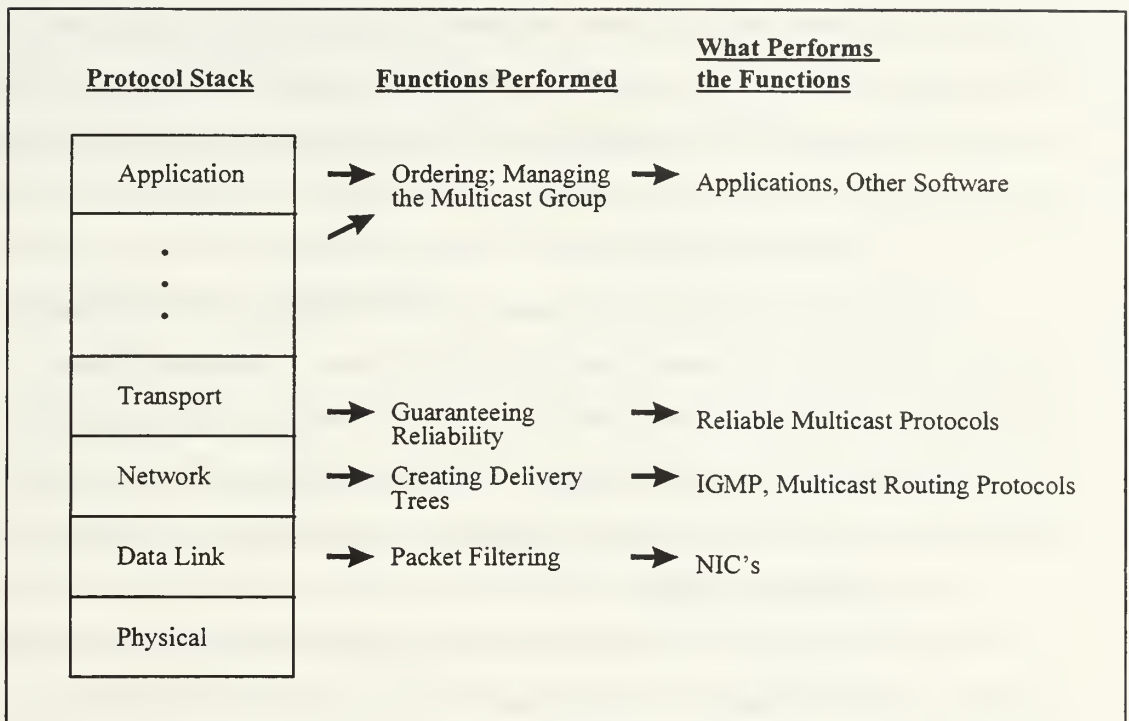
Figure 1.4. How Different Layers of the Protocol Stack Contribute to Multicasting.

2.   Data Link Layer

The two functions essential to multicasting that are performed at the data link layer relate to group addressing. First, protocols in this layer distinguish between group addresses and other addresses. The Ethernet, Token Ring, Token Bus, and Fiber Distributed Data Interface (FDDI) Media Access Control (MAC) protocols each provide some means of indicating if an address is a group address (Dempsey, 1990). Further explanation of how the group address is used by multicast protocols can be found in Chapter III.

Second, if a host belongs to a multicast group, the Network Interface Card (NIC) will deliver MAC frames with the appropriate group address to the host. Filtering packets with the NIC is more efficient than filtering them with higher layer software since "…doing it in software is orders of magnitude slower." (Dempsey, 1990)

Since radio networks share a common medium, access should be regulated by a MAC protocol. However, the MAC protocols designed for local area networks assume that all hosts can "hear" attempts to access the shared medium. Because the same

assumption is not valid for radio networks, MAC protocols which are tailored to the unique requirements of this environment need to be developed. Although an important component of tactical data communications, MAC protocols are beyond the scope of this thesis.

3. Network Layer

Two types of network layer protocols perform complimentary multicast functions. The Internet Group Management Protocol (IGMP) informs a host's immediately neighboring router of its desire to join or leave a multicast group. Multicast routing protocols build delivery paths from a sender to routers with group members on their attached networks. Both of these types of protocols are the subject of Chapter III.

4. Transport Layer

Reliable transport layer multicast protocols guarantee the reliable delivery of data multicast over the delivery trees built by network layer multicast routing protocols. Reliable delivery means that data arrives at all the intended receivers as it was sent by the source. In some ways multicast protocols at this layer are no different than their unicast counterparts. Like unicast protocols, they are responsible for error control, flow and rate control, and connection establishment and termination. The functions they may perform which have no unicast equivalent are guaranteeing certain types of ordering, and managing members of the multicast group.

Chapter IV discusses the design choices made by the developers of reliable transport layer multicast protocols. Seven of these protocols are evaluated in Chapter V.

Because reliable transport layer multicast protocols distribute data over delivery trees built at the network layer, multicast routing protocols will also be evaluated for their ability to operate within the constraints of the tactical internet (Chapter III).

5. Higher Layers

For some protocols, certain aspects of reliability are handled by layers higher than the transport layer. Ordering and group management, for example, may be the responsibility of higher layers. Some protocols also involve higher layers in failure recovery and flow control.

Where a function resides in the abstraction of the ISO reference model does not necessarily determine where it is best implemented. Arguments about where reliability functions should be implemented are presented in Chapter IV. The protocols in Chapter V represent a broad spectrum of design choices, with some confining reliability functions to the transport layer while others relegate them to higher layers.

D. THE TACTICAL INTERNET

1. Organization

The tactical internet will be organized to reflect the structure of the deployed force. The majority of hosts will reside on local area networks (LAN's) attached to routers; few will be connected directly to routers. LAN's connected to the tactical communications mesh by their attached routers will form a "network of networks." Although the types of point-to-point connections in this internet will be a function of the level in the organizational hierarchy, most wide area links will be provided by satellite or radio. The next chapter discusses the characteristics of the Marine Corps' tactical internet in greater detail.

2. Network Layer Protocols

Applications that will be hosted on the Marine Corps' tactical internet are based on the Transmission Control Protocol/Internet Protocol (TCP/IP). Currently, IPv4 is deployed; Nierle (1996) has recommended migrating to IPv6. Both versions support multicasting.

When a host informs its immediately neighboring router that it wants to join a multicast group, it specifies an IPv4 group address. Datagrams carrying a group address are delivered to the set of receivers after traversing the tree built by network level multicast routing protocols. This combination of the IPv4 group address, IGMP, and network level multicast routing protocols working together to multicast datagrams is often referred to as IP Multicast or Multicast IP.

Since the tactical internet will support multicasting at the network layer with Multicast IP, only those reliable transport layer protocols which run on top of Multicast IP will be considered in this thesis.

3. Characteristics of the Tactical Internet

Characteristics of the tactical internet will constrain the choice of a multicast protocol at both the network and transport layers. The tactical internet differs from the commercial internet in several respects. Bandwidth will be limited over most wide area point-to-point links by the transmission medium. Radio links, satellite links, and even some terrestrial wired links will not support the high data rates common in a commercial environment. Because some links will support higher capacities, the tactical internet will have to contend with the simultaneous presence of high and low bandwidth segments. Significantly higher bit error rates can also be expected over most links. Every part of the tactical internet will be subject to intentional disruption or destruction, with those parts of the internet in forward areas being at higher risk. The network topology can also be expected to change more frequently than commercial internets since units may relocate often or be reassigned; some users may constantly be on the move.

II. TACTICAL DATA NETWORK (TDN)

A. INTRODUCTION

To substantiate some of the generalizations made earlier about the characteristics of the tactical internet, this chapter describes the system that will form an integrated data network for Marine Corps tactical data systems.

Marine Corps organizational terminology will be explained before those aspects of the Tactical Data Network (TDN) related to multicasting are discussed. Included in this discussion will be characteristics of TDN communication links, as well as those tactical data systems likely to benefit from multicasting. The chapter will conclude with a summary of the results from a TDN simulation.

B. ORGANIZATION OF THE MARINE CORPS

During conflicts, the Marine Corps forms self-sustaining fighting organizations whose size is dictated by their assigned task. These task organized units are called Marine Air Ground Task Forces (MAGTF, pronounced "mag-taff"). Each consists of a ground combat element (GCE), air combat element (ACE), combat service support element (CSSE), and command element (CE). Infantry, artillery, and armor units make up the GCE; fixed and rotary wing aircraft and their supporting units comprise the ACE; logistics, engineer, communication and other support units fall under the CSSE; the commander and his staff are the command element. The ground, air, and combat service support elements of a MAGTF are often referred to as major subordinate commands (MSC's). See Figure 2.1.

```
                    ┌─────────────────────────┐
                    │    COMMAND ELEMENT      │
                    └─────────────────────────┘
```
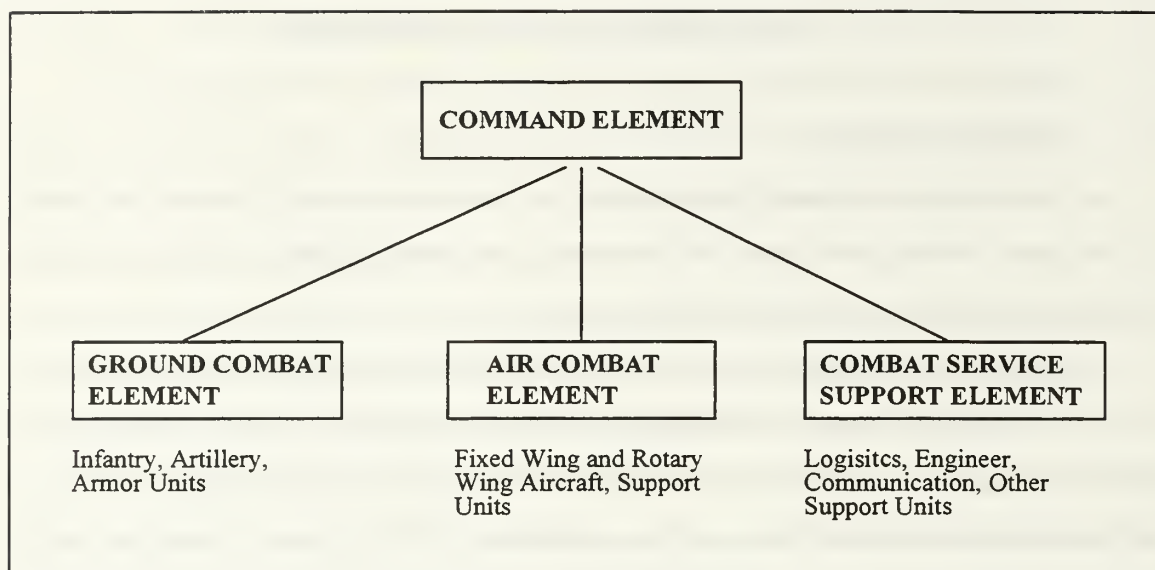
Figure 2.1. Marine Air Ground Task Force (MAGTF) Structure.

The largest task force that can be organized is called a Marine Expeditionary Force (MEF). The units which flesh out each element of the MEF are the largest of their type. The MEF GCE is staffed by a Division, the largest combat unit in the Marine Corps. Each Division consists of the infantry, artillery, and armor units mentioned earlier. Similarly, the ACE is formed by a Wing, while a Force Service Support Group (FSSG) makes up the CSSE. The appropriate size staff commands the MEF. For conflicts which require more lethality than a single MEF, more than one MEF can be deployed.

Smaller sized MAGTF's can be formed by piecing together elements further down in the organizational hierarchy. For example, the next smaller MAGTF is a Marine Expeditionary Brigade (MEB), comprised of units one rung lower than those which make up the MEF. The GCE of the MEB is filled by a Regiment, an Aircraft Group makes up the ACE, and a Brigade Service Support Group (BSSG) provides combat service support. The smallest MAGTF is a Marine Expeditionary Unit (MEU). Its warfighters are a Battalion of Marines; air support is provided by a Squadron; combat service support is provided by a MEU Service Support Group (MSSG). See Figure 2.2.
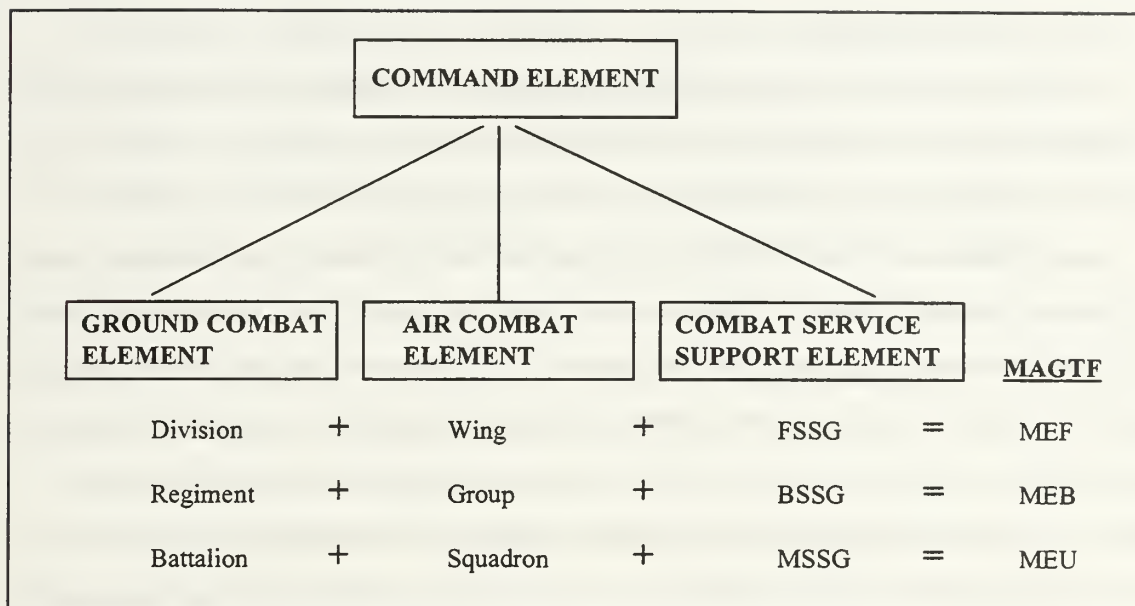
Figure 2.2. Composition of Different Sized MAGTF's.

C. TACTICAL DATA NETWORK (TDN)

1. System Description

TDN stitches together tactical data systems hosted on Ethernet local area networks. Server equipment will connect these LAN's to the existing tactical communications infrastructure. Gateways will join the tactical internet to external networks. (MARCORSYSCOM, 1995)

In addition to other equipment, the Server suite will include a multi-protocol router, and LAN hubs. Each Server can support an Ethernet LAN with 48 users, although segments can be bridged together to form multi-segmented LAN's. One or more Servers will be deployed to each unit down to the Battalion/Squadron level. (MARCORSYSCOM, 1995)

Gateways will be mounted in heavy variant High Mobility Multi-Wheeled Vehicles (HMMWV's). The multi-protocol routers, and LAN hubs of the Gateway equipment suite are augmented with patch panels, cables, and other equipment needed to connect to external communications systems. Gateways will allow TDN to "...interoperate with other service's tactical packet switching systems, as well as with strategic level networks." (Nierle, 1996) Each MEF will be supplied with two Gateways, and four Servers; each major subordinate command of the MEF will receive one Gateway and four Servers. (MARCORSYSCOM, 1995)

2. Connectivity

Tactical connectivity is primarily hierarchical since it reflects the chain of command. However, lateral links may be installed to improve traffic flow, avoid bottlenecks, and enhance reliability and survivability. (MARCORSYSCOM, 1995)

The deployed MAGTF will normally connect out of theater to the Defense Information Systems Network (DISN) by satellite. "The limited capacity of these links to DISN has always been and is expected to remain a significant choke point for deployed MAGTF's." (MARCORSYSCOM, 1995) Links between the MEF command element and its MSC's, the Joint Task Force (JTF) headquarters, and other Service Component headquarters will also be provided primarily by satellite, although multichannel radio may be used. The Division, Wing, and FSSG will communicate laterally with each other and with their immediately subordinate units via multichannel radio. (MARCORSYSCOM, 1995) The connectivity expected at the MEF by 1998 is shown in Figure 2.3.
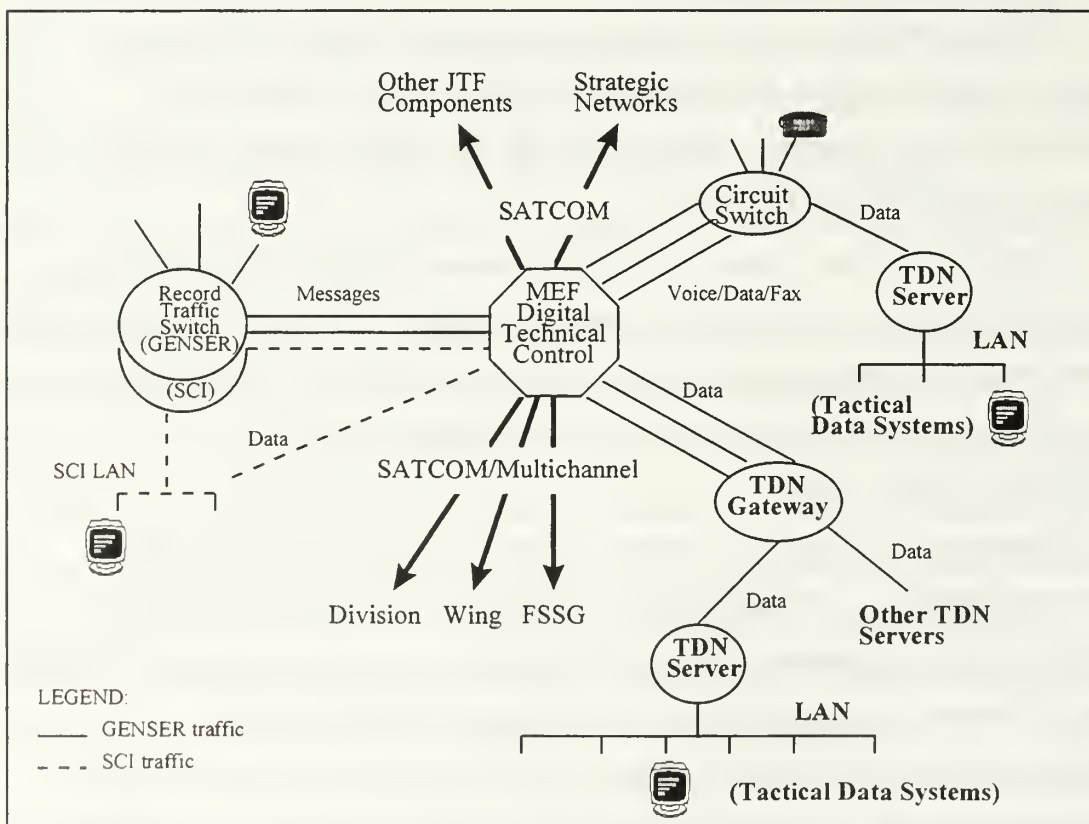
Figure 2.3. Connectivity Expected at the MEF by 1998. After (MARCORSYSCOM, 1995).

"The digital communications backbone of multichannel radio links stops at the Regimental/Group level, with just few exceptions." (MARCORSYSCOM, 1995) Even so, the transmission rate for circuits which feed into this digital backbone is limited to 16 or 32 Kbps by the equipment which provides access. "There is currently no plan to change this situation." (MARCORSYSCOM, 1995) Point-to-point links at levels lower than the Regiment will rely heavily on single channel radio. Although the single channel radios are also digital, they were designed primarily for the lower speeds (16 Kbps) required by voice traffic. (MARCORSYSCOM, 1995)

3. Tactical Data Systems

Some of the applications hosted on TDN which may benefit from reliable multicasting are described below.

a. *Tactical Combat Operations (TCO)*

"TCO will be the primary tactical data system used by commanders and operations officers...." (Nierle, 1996) By combining information about the enemy with the known status of friendly forces, TCO will provide commanders with a comprehensive, near real-time view of the battlespace. The updates needed to maintain a common tactical picture among many users may best be distributed by a reliable multicast protocol.

TCO will generate the most traffic of any of the data systems hosted on the tactical internet. This volume is a function both of the large number of users, and the types of applications included in TCO. (Nierle, 1996) Video teleconferencing is planned for later phases of its development; distributed collaborative planning may be an added capability. For either type of application, multicast protocols will reduce the unnecessary traffic generated by the unicast protocols of TDN.

b. *Intelligence Analysis System (IAS)*

"IAS provides automated support for the direction, collection, processing, production, and dissemination of intelligence within the MAGTF." (Nierle, 1996) Intelligence may either be distributed to multiple recipients without their explicit request ("push"), or consumers may download needed information after browsing through a centralized intelligence repository ("pull"). In both scenarios, the reliable transfer of data

is necessary. In the first case, the data must be delivered reliably to more than one recipient; for intelligence "pull", the centralized repository may have to be replicated at other sites. Reliable multicast protocols can perform both types of operations more efficiently than their unicast counterparts.

c. *Advanced Field Artillery Tactical Data System (AFATDS)*

AFATDS is an automated command and control system for artillery units and the units that coordinate artillery fires. (Nierle, 1996) Some reliable multicast protocols would be well suited for distributing the critical, real-time AFATDS data to those providing and coordinating fire support.

d. *Marine Combat Service Support Command and Control System (MCSSC2)*

Although tailored for combat service support units, MCSSC2 is similar to other command and control systems. Data will be aggregated from many sources in the CSSE and processed into a coherent representation that will be shared by decision makers. Information will also flow downward and laterally in the CSSE organization from one or more senders to many recipients. The precise nature of logistics support demands reliable data delivery; the one-to-many communications is a good fit for multicasting.

D. SIMULATION OF THE TACTICAL INTERNET

1. Description

The MITRE Corporation simulated the network traffic generated by four tactical data systems for each of the components of a MEF. The data systems included in the simulation were AFATDS, IAS, MCSSC2, and TCO. The measures of performance were transmission delay, and channel utilization capacity. (Deichler, 1996)

Because no representative data was available for the frequency or size of file transfers for these systems, only Marine Tactical Systems (MTS) messages were generated for each system. Even so, the worst case was simulated by setting all messages to their maximum size. E-mail traffic was not simulated; neither was traffic from units at the Battalion level and below included in the study. (Deichler, 1996)

The protocols supported by the simulator included Ethernet, Token Ring, TCP/IP, Integrated Systems Digital Network (ISDN), and FDDI; multicast protocols were not supported. (Deichler, 1996)

2. Results

The results from the GCE Division LAN and the GCE Gateway are representative of the mismatch in capacity between LAN's and wide area point-to-point links. As expected, very little (< .5%) of the GCE Division LAN capacity was used, even when the baseline traffic load was increased by a factor of 35. The same loading consumed almost 23% of the capacity of the GCE Gateway link. (Deichler, 1996)

E. CONCLUSIONS

Many of the data systems hosted on the tactical internet will require reliable one-to-many communications which do not overload its low speed point-to-point links. Reliable multicast protocols are able to deliver the functionality needed by tactical data systems while remaining stingy with bandwidth.

Not only will bandwidth be limited in the tactical internet, but higher bit error rates can be expected over most links. Burst errors are common over satellite links; all radio links are susceptible to both intentional and unintentional interference.

Those multicast protocols best suited for the tactical internet will most closely meet the requirements imposed by the tactical environment.

III. MULTICAST ROUTING PROTOCOLS

A. INTRODUCTION

For hosts to multicast data across an internet they must have some means of informing routers of their membership in a multicast group, and the routers must know where to send the data so that it is delivered from the source to all members of the group. Hosts use the Internet Group Management Protocol (IGMP) to communicate with their immediately neighboring router; the multicast routing protocols discussed throughout this chapter help routers establish a delivery path through the internetwork from a multicast source to the members of a multicast group.

The primary intent of this chapter is to determine how well suited each routing protocol is for constructing multicast delivery trees in the tactical internet. Before each protocol is examined, two other important pieces of the network layer multicast puzzle, multicast group addressing and IGMP, will be discussed. This chapter will conclude with an explanation of the experimental results from a comparison of several protocols.

B. MULTICAST IP

1. Multicast Addresses

All receivers in the same multicast group are identified by one multicast address. Of Class D, multicast addresses are identified by "1110" in the most significant positions of the IP address, and range from 224.0.0.0 - 239.255.255.255. To participate in multicasting, hosts and routers must be able to process Class D addresses.

Unlike unicast IP addresses, the Class D multicast address structure is flat. Whereas unicast addresses assigned to a host at a site must be drawn from an address space which may be topographically or geographically assigned, or dependent on the Internet provider to that site, multicast addresses have no such association (Crowcroft, 1996). Voigt (1996) feels that this flat address space is a liability since any multicast could potentially become global in scope.

According to Nierle (1996), the Marine Corps "...must adopt some consistent means of obtaining and managing multicast addresses." When Class D addresses are assigned dynamically, hosts learn of a new group address only after the first member of

the group has announced it to the network. Some hosts may not receive the announcement if it is lost, or if its scope is limited. "Passive procedures such as this do not fit with the changing nature of tactical topology and the users' needs to rapidly exchange and distribute information." (Nierle, 1996) Nierle suggests assigning well-known, semi-permanent multicast addresses to "...high priority groups, and to multicast groups that develop naturally from doctrinal command structure." (Nierle, 1996)

2.   IGMP

IGMP allows hosts to join and leave multicast groups. By sending a Membership Report to its immediately neighboring router, a host informs the router that it wishes to become part of a multicast group. Routers periodically transmit Membership Query messages to determine which host groups have members on their directly attached networks (Semeria, 1996). These query messages are addressed to all hosts and have a time to live (TTL) of 1 to limit their transmission to the network directly attached to the router. A host responds with a Membership Report for each group to which it belongs. To limit the number of Membership Reports, each host begins a random delay timer upon receiving a Membership Query. Hosts "listen" to the Membership Reports sent to the router; if a report is submitted for the group to which a host belongs before its timer expires, it cancels its report for the group. This mechanism ensures that only one membership report is generated for each group. Based on the group membership information gained through IGMP, routers are able to determine what multicast traffic (if any) to forward to its attached networks (Semeria, 1996).

When application software tells the host networking software to join a multicast group, an IGMP message is sent to the host's neighboring router (if the host is not already a member of the group). At the same time, the Class D multicast address of the group to be joined is mapped to a lower level address and the network interface is programmed to accept packets to this address. (Crowcroft, 1996) For example, if a host joins a group on an Ethernet interface, the lower 23 bits of the Class D address are mapped to the lower 23 bits of the Ethernet address. Because of this hardware "filtering" of multicast addresses, a router does not need to maintain a detailed list of which hosts belong to each multicast

group, but only that at least one member of the group is present on its attached network (Semeria, 1996).

One of the weaknesses of the first version of IGMP was the high latency associated with terminating multicast sessions. After the last member of a multicast group on a subnetwork had left a group, other routers were not immediately notified to stop forwarding traffic for the group. This delay was caused by IGMP waiting until several queries indicated that no members of a particular group remained on the subnetwork. In the meantime, though, unnecessary traffic would be forwarded to the subnetwork. The cost of this unnecessary traffic could be high, particularly in a bandwidth constrained tactical internet.

IGMP Version 2 introduces some refinements which will help reduce protocol overhead. The Group Specific Query Message allows routers to query specific groups on their directly attached networks instead of being forced to query all groups. Beginning with version 2, the termination of a multicast session is no longer done passively. The last host on a subnetwork to leave a multicast group transmits a Leave Group message to the router which indicates the group to be left. After verifying the departure with a Group Specific Query Message, the router notifies other routers to stop forwarding traffic to the subnetwork for that group.

IGMP Version 3, which was a preliminary draft specification in March 1996, goes further in reducing protocol overhead. Bandwidth will be conserved by the Group-Source Report message which allows hosts to receive traffic from specific sources of a multicast group. In previous versions of IGMP, traffic from all sources had to be forwarded to a subnetwork even if hosts were only interested in receiving traffic from specific sources. The Leave-Group messages first introduced in Version 2 have also been enhanced to allow hosts to leave an entire group or to specify the specific source they wish to leave (Semeria, 1996).

Because later versions of IGMP can reduce unnecessary traffic, they should be favored over Version 1 for installation on hosts and routers in the tactical internet.

C. DELIVERY TREES

1. Introduction

Multicast routing protocols build delivery trees which are either source-specific or center-specific. The Distance Vector Multicast Routing Protocol (DVMRP), Multicast Extensions to Open Shortest Path First (MOSPF), and Protocol Independent Multicast-Dense Mode (PIM-DM) build source-specific trees; Center Based Trees (CBT) and Protocol Independent Multicast-Sparse Mode (PIM-SM) construct center-specific trees.

2. Source-specific Trees

To route multicast traffic through an internet, routers could employ a flooding algorithm. Upon the arrival of a multicast datagram, a router would determine if it had seen the packet before. Duplicates would be discarded, while a new packet would be forwarded on all router interfaces except the one on which it had arrived. While such a scheme would be simple to implement, it would waste bandwidth. Router memory would also be used inefficiently since a table entry would have to be kept for each recently seen packet. (Semeria, 1996)

A more effective method of delivering multicast traffic is for the routing protocol to build a spanning tree. A spanning tree "…defines a tree structure where only one active path connects any two routers." (Semeria, 1996) For traffic to traverse a spanning tree, routers forward a multicast packet on all interfaces that are part of the spanning tree except the interface on which it arrived. Several routing protocols construct spanning trees rooted at the source of a multicast group. "Since there are many potential sources for a group, a different spanning tree is constructed for each active (source, group) pair." (Semeria, 1996) The widespread use of spanning tree protocols in bridges suggests that spanning trees are a well understood delivery mechanism. Figure 3.1 depicts the spanning trees for two sources belonging to the same multicast group.

20
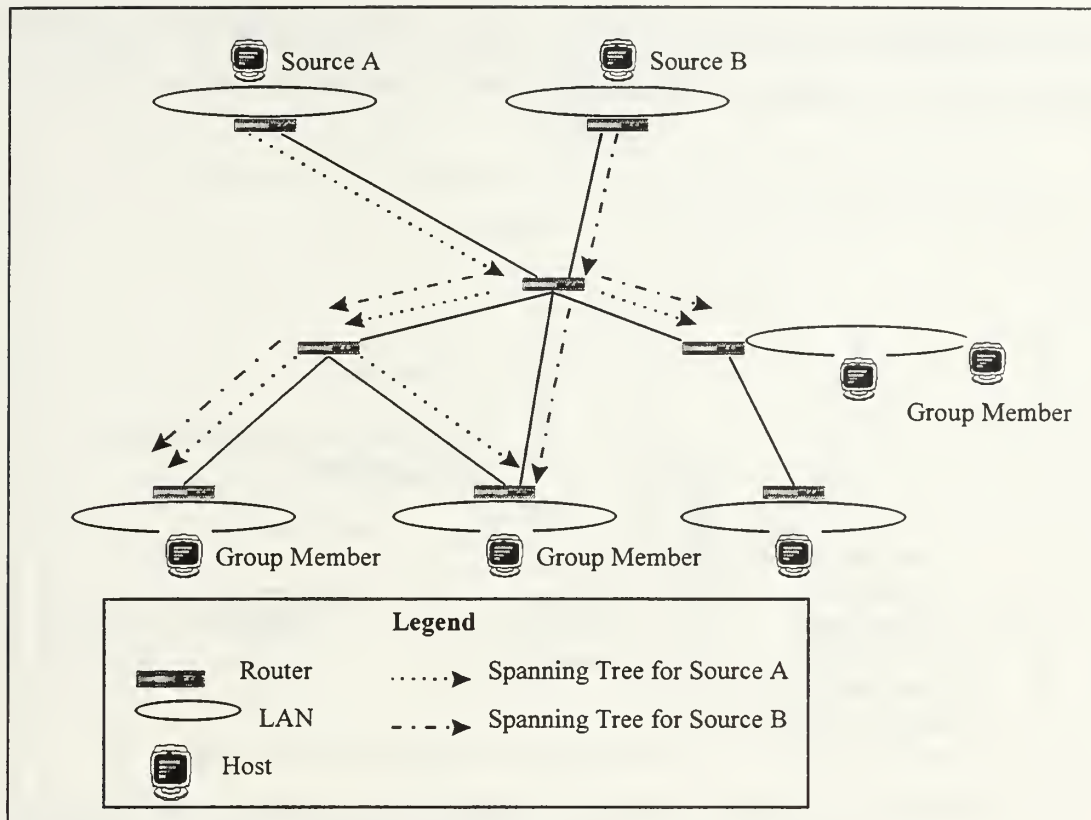
Figure 3.1. Source-Rooted Spanning Trees

The three algorithms used by routing protocols to build source rooted spanning trees are Reverse Path Broadcasting (RPB), Truncated Reverse Path Broadcasting (TRPB), and Reverse Path Multicasting (RPM).

a. Reverse Path Broadcasting (RPB)

Reverse Path Broadcasting is the algorithm upon which the other two variants are based. Some authors (Tanenbaum, 1996; Voigt, 1996) also refer to this algorithm as Reverse Path Forwarding (RPF). According to the RPB algorithm, packets that arrive on an interface considered to be the shortest path back to the source are forwarded on all interfaces except the incoming interface; otherwise, the arriving packet is discarded. (Semeria, 1996)

The shortest path link back to the source of a particular multicast is termed the "parent" link of a router. Links over which packets can be forwarded are called "child" links. (Semeria, 1996) The number of packets forwarded by RPB can be reduced if packets are only forwarded on child links to routers which consider the router making the forwarding decision to be on its shortest path back to the source. Packets forwarded

21

on other child links are superfluous since they will be discarded by downstream routers. (Semeria, 1996) The operation of the enhanced RPB algorithm is illustrated in Figure 3.2.
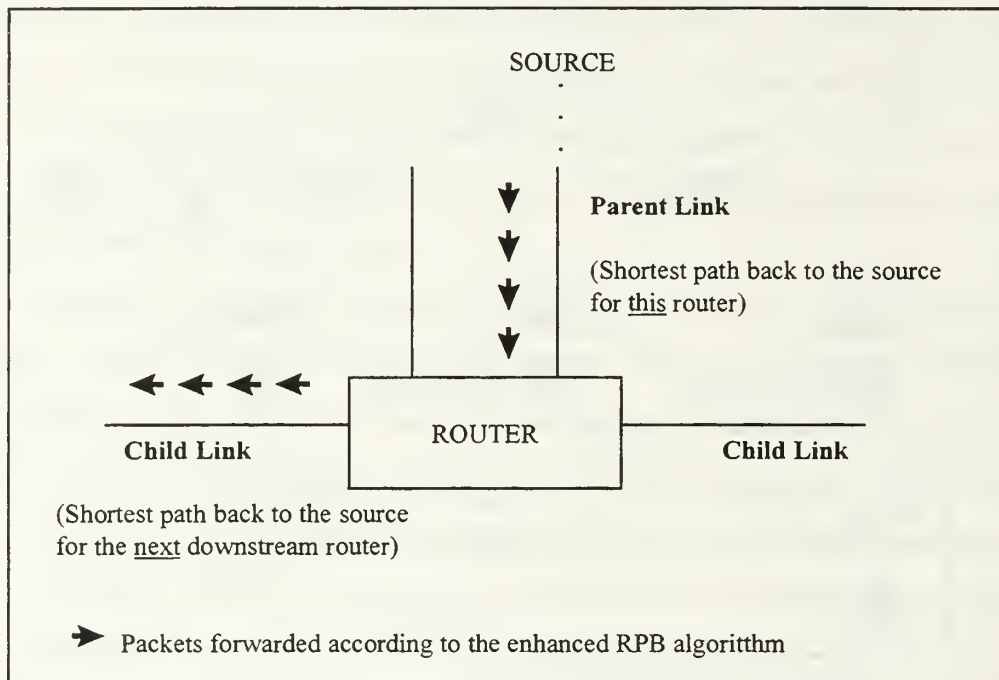
Figure 3.2. RPB Forwarding Algorithm. After (Semeria, 1996).

The principle advantage of RPB is that the algorithm is both reasonably efficient and easy to implement (Tanenbaum, 1996). Routers do not need to know about the entire spanning tree; only the shortest path distance between nodes is required (Voigt, 1996). Furthermore, trees built using RPB are the shortest path from the source to the receiver group. Finally, when trees are built for each source to a multicast group, traffic is distributed over multiple links resulting in better network utilization (Semeria, 1996).

The disadvantage of RPB is that in a multicast environment, where only a subset of the hosts are members of the receiver group, packets will be delivered to nonmembers (Voigt, 1996).

b. Truncated Reverse Path Broadcasting (TRPB)

From the group membership information provided to routers by IGMP, the Truncated Reverse Path Broadcasting algorithm stops traffic from being forwarded onto a LAN without any members of the destination group. A LAN that has been truncated from the multicast delivery tree by TRPB is shown in Figure 3.3.

22
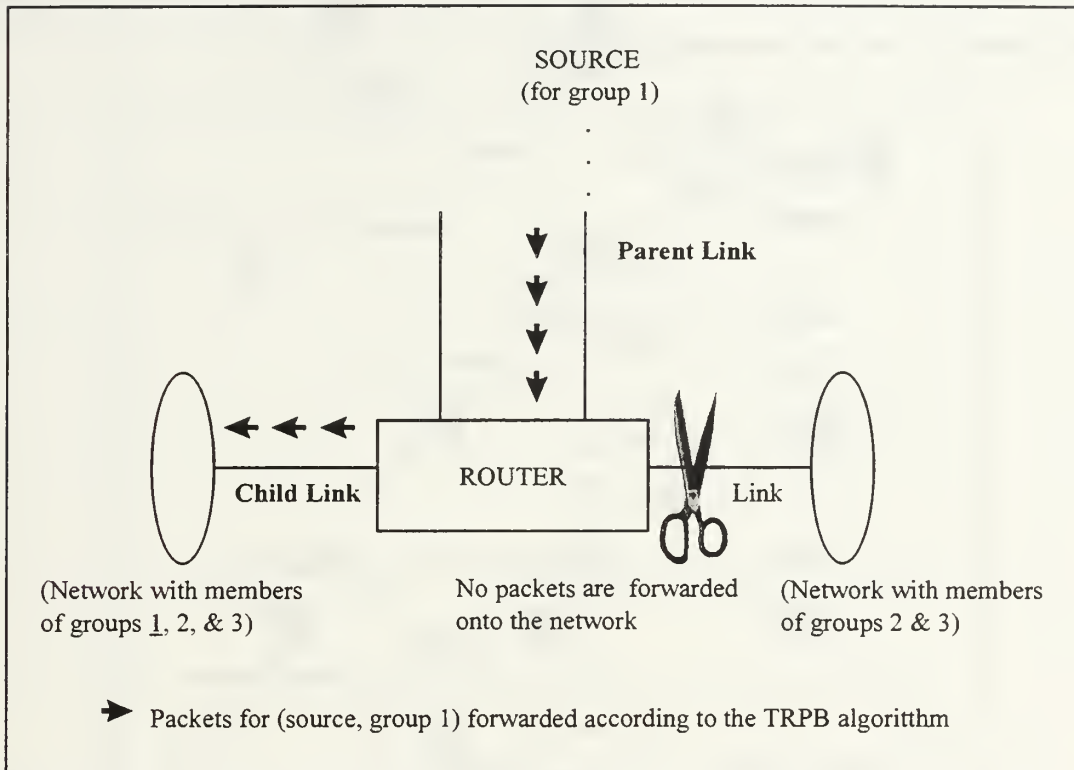
Figure 3.3. TRPB Forwarding Algorithm. After (Semeria, 1996).

Although TRPB eliminates unnecessary traffic to LAN's at the edge of the network, hosts which are not part of the multicast group still receive packets since TRPB "...does not consider group memberships when building the branches of a distribution tree." (Semeria, 1996)

c.   *Reverse Path Multicasting (RPM)*

Reverse Path Multicasting improves upon TRPB by creating a delivery tree that "...spans only subnetworks with group members, and routers and subnetworks along the shortest path to subnetworks with group members." (Semeria, 1996)

Tree construction is begun with a packet forwarded using TRPB. A LAN without members of the destination group sends a "prune" message one hop towards the source over its parent link. Likewise, an intermediate router which receives prune messages on all its child links sends a prune message one hop back up the tree towards the source. The resultant shortest path tree contains only routers attached to LAN's with members of the destination group and routers on the path to routers which have members attached. (Semeria, 1996) A tree pruned by RPM is shown in Figure 3.4. Had the

23

source-rooted tree in Figure 3.4 been built with the TRPB algorithm, packets would have been forwarded over the links pruned by RPM.
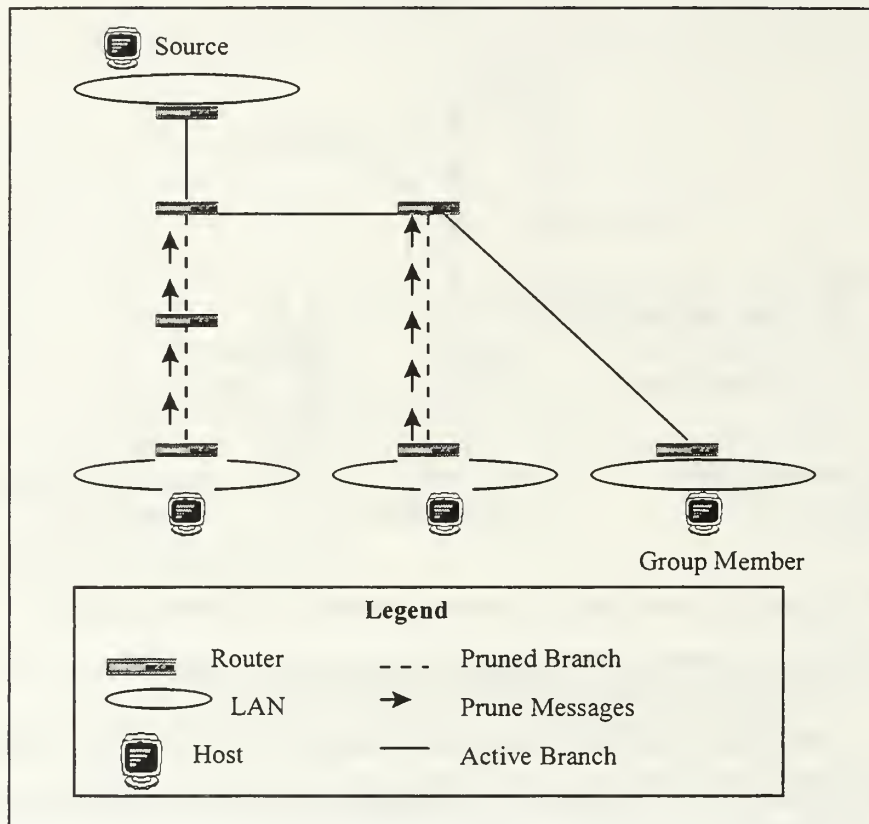
Figure 3.4. RPM Forwarding Algorithm. After (Semeria, 1996).

By periodically reconstructing the multicast delivery tree, RPM is able to adapt to changing group membership and network topology. However, the flood and prune strategy which makes RPM adaptive is also its biggest limitation because of the bandwidth it consumes. Another disadvantage of the algorithm is that each router must maintain state for each (source, group) pair. (Semeria, 1996)

3. Center-specific Trees

Instead of building a shortest path tree for each source of a multicast group, routing protocols such as CBT and PIM-SM build a single tree rooted at a central router or group of routers. For center-specific trees "...multicast traffic for each group is sent and received over the same delivery tree, regardless of the source." (Semeria, 1996) An example center-specific tree can be found in Figure 3.5.
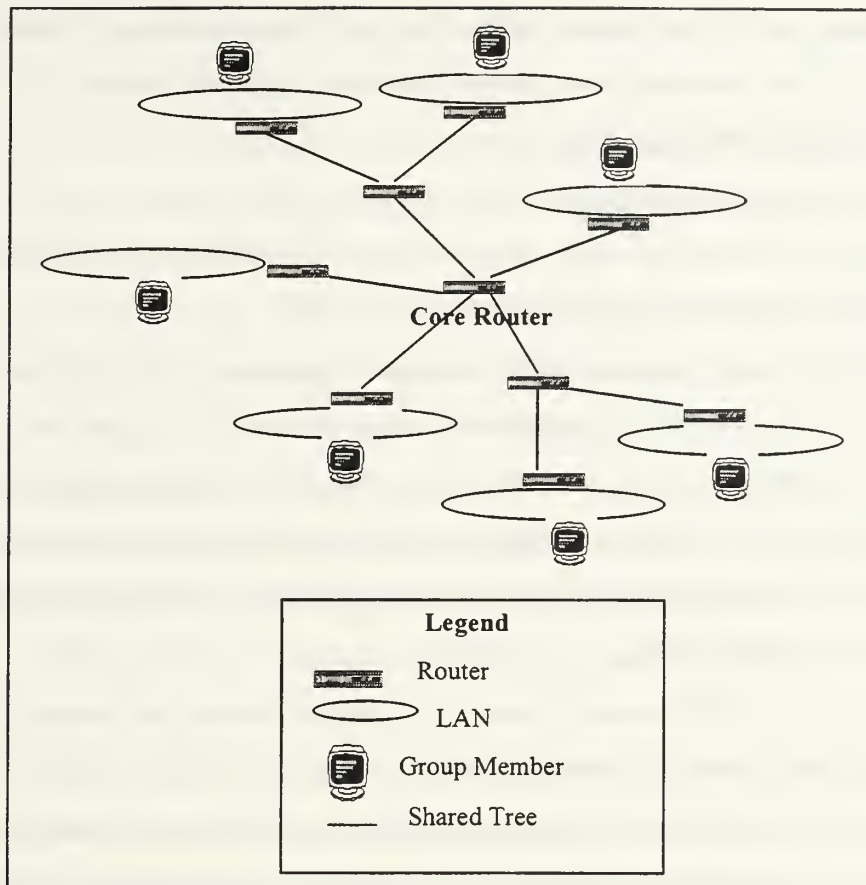
Figure 3.5. Center-specific Tree

To join a multicast group, a host sends a unicast message towards the core of the shared tree. Intermediate routers mark the interface on which the join request was received as belonging to the delivery tree and forward the request until it is received by one of the core routers. Sources transmit to the destination group by multicasting packets towards the core. The core then forwards the packets over its outgoing interfaces to the remaining branches of the shared delivery tree.

The most significant advantage offered by center-specific trees is the reduced amount of bandwidth they consume since trees are constructed without the costly flood and prune algorithms of protocols which build source-specific trees. Another benefit of this approach is that routers which participate in building center-specific trees only maintain state for each group instead of for each (source, group) pair. (Semeria, 1996)

However, there are drawbacks to center-specific multicast delivery trees. The high connectivity of the core routers may result in traffic concentrations and bottlenecks. Also, unlike the shortest path trees built with source-specific algorithms, center-specific

25

trees may create paths which are not optimal for some group members. Finally, workable algorithms for core placement have yet to be developed. (Semeria, 1996)

D. ROUTING PROTOCOLS

1. Evaluation Criteria

The following criteria can be used to evaluate the multicast routing protocols which will be discussed in the remainder of this chapter:

- state - costs to routers, which include: (1) memory required for forwarding tables and configuration information (such as neighbors, and core membership), and (2) the costs associated with calculating this information (Crowcroft, 1996). For multicast delivery, state may be required in routers which are not part of the delivery tree ("off tree"), as well as in "on tree" routers (Xu, 1996).

- control traffic overhead - packets exchanged to set up and maintain multicast delivery trees (Estrin notes, 1996). The amount of traffic which both "on tree" and "off tree" routers are required to process should be considered. Control traffic overhead affects scalability and is an important consideration in a tactical internet.

- data distribution overhead - overhead associated with distributing data packets (Xu, 1996).

- scaling - how well a protocol adjusts to an increase in the number of senders or receivers. Factors affecting how well a protocol scales include "...network state maintenance, bandwidth efficiency, and protocol overhead." (Ballardie, 1996) These factors, in turn, can be affected by the number of senders and the distribution of the destination group.

- join latency - the amount of time it takes for a host to begin receiving data for a group after sending an IGMP Membership Report to its immediately neighboring router (Estrin notes, 1996).

- complexity of implementation - a subjective measure of the difficulty of implementing the protocol. The complexity of a routing protocol may be a

26

factor in the amount of effort required to operate and maintain a network. (Estrin notes, 1996)

- delay - delay from a source to all recipients of a destination group.
- link reutilization - the amount of traffic concentrated on a given link (Crowcroft, 1996).
- convergence time - the time it takes for the network to adjust to a change in topology. Convergence time is an important measure, especially for protocols operating in the dynamic tactical environment. However, a lack of data makes substantive comparisons of protocol convergence difficult.

The results of experiments conducted to compare several of the protocols using some of the above metrics will be discussed at the end of this chapter.

2. Distance Vector Multicast Routing Protocol (DVMRP)

a. Protocol Overview

Like distance vector routing protocols, DVMRP exchanges multicast routing table update messages with its neighboring routers, but only if they are multicast capable. Unicast updates are sent separately, and different processes must be run to route multicast and unicast traffic. (Semeria, 1996)

DVMRP uses the RPM algorithm to build its multicast delivery trees. After a specified interval, pruned branches grow back and another datagram must be forwarded to redefine the delivery tree. To reduce latency for hosts that join a group between these refresh intervals, DVMRP employs a "graft" mechanism which restores previously pruned branches to the tree. (Semeria, 1996)

The routing table built from exchanged updates represents the "…shortest path source-rooted spanning tree to every participating subnetwork". (Semeria, 1996) The forwarding table, which is derived from the routing table, known groups, and received prune messages, identifies those interfaces on which traffic must be forwarded for it to be delivered over the RPM tree for each (source, group) pair. (Semeria, 1996)

b. Evaluation

The trees built by DVMRP offer the shortest hop count distance from each source to its receivers. However, the periodic flood and prune RPM algorithm which

DVMRP uses to build its multicast delivery trees limits the scalability of the protocol, especially in a sparse environment. During the flood stage, control traffic reaches both "off tree" and "on tree" routers. Another scaling problem is that changes in topology are propagated slowly throughout the network since routers only update their immediate neighbors during each update period (Crowcroft, 1996). Also, routers not participating in the multicast delivery for a group are still required to maintain a routing table for the RPB tree (which reaches all subnetworks). Finally, data distribution overhead is high since it includes the overhead incurred by the data packet sent each time the tree is periodically refreshed.

3.   **Hierarchical Distance Vector Multicast Routing Protocol (HDVMRP)**

DVMRP is the protocol used by most multicast routers (mrouters) in the rapidly growing MBone. The increasing amount of state which each router must maintain as the MBone grows, as well as the scaling problems mentioned in the previous section, have led to the development of a hierarchical variant of DVMRP.

Instead of the current flat structure, the network will be divided into non-intersecting regions. Routing within a region can be performed by any multicast protocol (level 1); DVMRP will be used to route traffic between regions (level 2). (Semeria, 1996)

According to Semeria (1996), the advantages to such a hierarchical structure are: (1) different multicast routing protocols can be implemented in each region, (2) failures within a region are limited to a single domain, and (3) topological changes between regions are only a concern of inter-domain routers. Voigt (1996) however, views Hierarchical DVMRP as a "…specific incremental improvement to an existing protocol." He objects to the flooding which will still occur within those regions that deploy DVMRP, and between regions.

4.   **Multicast Extensions to Open Shortest Path First (MOSPF)**

a.   *Protocol Overview*

Link state routing protocols, such as the unicast OSPF, build router databases which describe the complete network topology. The multicast extension to OSPF adds an advertisement that communicates group membership. From these

28

membership updates and the network image it stores, each router is able to calculate source rooted shortest path multicast trees using Dijkstra's algorithm. (Semeria, 1996)

A single designated router (DR) on each subnetwork is responsible for learning of group membership through IGMP queries. The DR updates the group membership at other routers on the subnetwork by flooding group membership advertisements. The shortest path tree for a source is calculated when a router receives the first multicast packet for a particular (source, group) pair. Because this calculation is performed in memory, the first datagram does not have to be forwarded to all routers, as in DVMRP. After the shortest path tree has been calculated, and a router has determined its position in the tree, a cache is created to forward all subsequent packets for the (source, group) pair. Unlike DVMRP, the delivery tree is not periodically refreshed. A new tree is calculated only when the network topology or group membership changes. (Semeria, 1996)

MOSPF supports routing between different OSPF areas and between different autonomous systems. In both cases, specially designated routers, called multicast forwarders, receive all multicast traffic for an area before forwarding it onto the links of the wider topology. (Semeria, 1996)

b. *Evaluation*

The strength of MOSPF, like DVMRP, is that delay is minimized since traffic is delivered over shortest path trees. Also like DVMRP, the primary disadvantage of the protocol is its lack of scalability. Not only does the broadcasting of group membership advertisements limit scalability, but it results in control traffic being delivered to "off tree" as well as "on tree" routers. The amount of memory required by each router to store an image of the entire network further limits the scalability of the protocol. Another characteristic MOSPF shares with DVMRP is the high overhead associated with distributing data since the delivery tree is built when the first data packet is sent. The join latency is also high since, for each join, a shortest path tree has to be calculated and group membership advertisements have to be broadcast. Since topological changes also trigger costly recalculations of shortest path trees in all routers, MOSPF

might not be suited for dynamic network environments. The final disadvantage of this protocol is that it can only be deployed in networks running OSPF.

5.   **Core Based Trees (CBT)**

a.   *Protocol Overview*

A CBT shared tree is a "...set of pre-nominated cores (routers), to which routers connected to member hosts join by means of an explicit protocol control message." (Ballardie, 1996) The explicit join procedure is initiated when, through IGMP, a router learns of a host or hosts on its attached network wanting to join a multicast group. The join message sets up a transient branch of the shared tree as it is forwarded along the shortest path between the initiating router and the set of cores. The acknowledgment from the joined core, which actually creates a branch of the tree, is returned along the reverse path of the join request. Not all join requests travel to the set of cores, since CBT capable non-core routers can also acknowledge them. (Ballardie, 1996)

Information maintained about the shared tree is "hard state". A branch is created based on underlying unicast routing tables but subsequent changes in the tables do not necessarily prompt corresponding changes in the shared tree. Branches are torn down explicitly instead of timing out. The health of tree links and routers is monitored through periodic "keepalive" messages sent between adjacent routers. (Ballardie, 1996)

One shared tree is created for each group. Non-member sources can send to a group by addressing their traffic to the unicast address of the core. Once on the shared tree, the data is distributed to all members of the destination group. (Ballardie, 1993)

The principles of hierarchical PIM, which will be discussed in the next section, may also apply to CBT. (Ballardie, 1996)

b.   *Evaluation*

CBT was developed primarily to overcome the scaling problems of DVMRP and MOSPF. For source-specific multicast routing protocols, the amount of state maintained by each router is proportional to the number of sources multiplied by the number of groups. Since CBT constructs one shared tree for each group, router state is

proportional only to the number of groups, making it well suited for applications with many senders. An additional benefit of CBT is that state is stored only in "on tree" routers.

CBT builds its delivery trees from unicast routing tables, regardless of the protocol used to compute them. This protocol independence should make CBT easier to deploy, as should the fact that routers of non-member sources do not have to be multicast capable, since their traffic is unicast towards a core on the tree.

Control traffic is minimal. Trees are set up through explicit join requests, and maintained by periodic "keepalive" messages.

"The advantage of a hard state is that the tree is constructed and removed explicitly..." thereby eliminating the need for costly flooding and pruning algorithms. (Voigt, 1996) Explicit construction also implies that the first packet of a multicast transmission will not be delayed by construction of the delivery tree as in DVMRP and MOSPF. An added benefit of CBT hard state is that resources may be reserved as the delivery tree is set up (Ballardie, 1996); the drawback is that CBT may be unable to adapt to changing delay conditions in the network (Voigt, 1996).

Certain applications may not be appropriate for CBT since the delay of shared trees is greater than that of source-based trees (Voigt, 1996). Other weaknesses of the protocol relate to the core. Ballardie (1996) writes that the issues "...of core...selection, placement, and management are still under review" leaving administrative placement of the core as the only current alternative. The centralized structure of shared trees and prominent role of the core in building and removing branches may concentrate traffic in the core area. Finally, shared trees are not as robust as source-based trees since the core is a central point of failure. Survivability can be improved, however, by increasing the number of cores.

6.   **Protocol Independent Multicast-Dense Mode (PIM-DM)**

a.   Protocol Overview

PIM-DM, like DVMRP, is designed for environments where group members are relatively densely packed and bandwidth is plentiful. Both protocols allow

router table entries to time out (soft state); delivery trees are rebuilt using RPM. PIM-DM is also like DVMRP in that it incorporates a graft mechanism to reduce join latency.

However, the two protocols differ in several respects. While DVMRP has its own protocol for computing unicast routing information, PIM-DM (and SM) are "…not dependent upon mechanisms provided by any particular unicast routing protocol." (Semeria, 1996) DVMRP also attempts to reduce the number of packets it forwards by computing those child links which are on the shortest distance path back to a source. PIM-DM, on the other hand, just forwards packets on all downstream interfaces. Prune messages eliminate routers not leading to group members and routers with no members on their directly attached LAN's. Although additional overhead is created by this approach, the designers were willing "…to accept some duplicate packets in order to avoid being protocol dependent and avoid building a child parent database." (Estrin, 1996)

b. *Evaluation*

The protocol independence of PIM-DM is one of the advantages it has over DVMRP. However, the duplicate packets PIM-DM forwards to achieve this independence make it less suitable for a wide area environment than DVMRP. The amount of state which routers maintain should be less than DVMRP by the amount required to store the child parent database, but will still be more than CBT. Other criticisms of DVMRP regarding the high overhead for delivering the first data packet after a refresh, high control traffic overhead, and state being maintained in "off tree" routers also apply to PIM-DM.

7. Protocol Independent Multicast-Sparse Mode (PIM-SM)

a. *Protocol Overview*

A single router can run both PIM modes. Designed for environments in which group members are distributed and bandwidth is limited, PIM-SM complements the dense mode variant.

Like CBT, PIM-SM builds a shared multicast distribution tree for each group. The central router around which the tree is built is called a Rendezvous Point (RP) since it is here that receivers learn of sources, and sources learn of receivers. An

32

explicit request to join a group is unicast to the RP by the designated router on behalf of a host on its directly attached or downstream network. The forwarding cache entry created by intermediate routers constructs a shortest path branch of the shared tree. (Semeria, 1996)

A new source informs the group of its presence by unicasting a multicast packet encapsulated in a registration packet to the RP. The join message sent back in response builds a branch of the shared tree between the RP and the source. Subsequent data packets multicast towards the RP are distributed to all group members on the shared tree. (Semeria, 1996)

Because the RP-shared tree may introduce added delay, receivers have the option of switching to a shortest path tree. To build the shortest path tree, a designated router sends a join request towards a source. Once the designated router begins receiving packets directly from the source, a prune message is sent to the RP so that packets no longer arrive over the RP-shared tree (Deering, 1996).

For PIM-SM to interoperate with dense mode protocols such as DVMRP, PIM-SM builds delivery trees which funnel all PIM multicast packets to routers joining the PIM domain to other domains. (Deering, 1996)

b.   *Evaluation*

Semeria (1996) writes that PIM-SM "…requires routers to maintain a significant amount of state information to describe sources and groups." However, for RP-shared trees, state is maintained only by "on tree" routers. It is not clear, from the specification, whether switching to the shortest path tree will require routers on the shared tree to continue maintaining state.

Besides the increased amount of state maintained by PIM-SM routers, the only difference between the two protocols is the increased complexity of PIM-SM. In other respects, the evaluations of PIM-SM and CBT are similar.

8.   Hierarchical Protocol Independent Multicast (HPIM)

HPIM "…is a proposal to solve very specific problems with PIM, that of the advertisement of rendezvous points (RPs) to group members and mapping RPs to groups." (Voigt, 1996) Both RP advertising and mapping are considered to be

undesirable since they require application and end system involvement (Ballardie, 1996). Another problem corrected by HPIM is a direct consequence of PIM-SM's flat structure which sometimes forces a source to send to a distant RP in order for traffic to be delivered to local receivers.

In HPIM, the RP's are structured into a hierarchy of increasing scope. Ballardie (1996) expects that global reach will be attained in "...six or seven levels of hierarchy...." When a host joins a group, its request is passed up the hierarchy of RP's until it attains the maximum level for the group or it reaches a router that has already joined the group (Crowcroft, 1996). Like PIM-SM, the first multicast packet from a new source is encapsulated in a registration packet. This packet travels up the hierarchy checking for receivers at each level. Subsequent data packets are sent unencapsulated to RP's which then multicast them to receivers in their level of the hierarchy.

Although Ballardie (1996) thinks that the principles of HPIM might also apply to CBT, Voigt (1996) characterizes HPIM as a "...specific incremental improvement to an existing protocol..." which "...introduces a new level of complexity and a new set of problems."

9.   Quantitative Comparisons

a.   Introduction

The quantitative results presented in this section have been included to lend credence to the protocol evaluations which were conducted earlier. Both studies discussed here compare the performance and overhead of center-specific trees to source-specific trees. The Harris Corporation study, found in (Ballardie, 1996), assessed the suitability of PIM and CBT for a Distributed Interactive Simulation (DIS) environment. PIM-DM, and PIM-SM with shortest path trees (SPT's) represented source-specific trees; CBT and PIM-SM were used as center-specific trees. Wei and Estrin (Wei, 1994) simulated random networks under different circumstances to compare source-specific and center-specific tree algorithms.

b.   Harris Corporation Study

The criteria used by Harris to evaluate the protocols were: end-to-end delay, group join time, scalability (all participants were both senders and receivers),

bandwidth utilization, overhead traffic, and protocol complexity (Ballardie, 1996). The simulation results, summarized in (Ballardie, 1996) are repeated here:

- End-to-end delay: PIM-DM and PIM-SM with SPT's deliver packets between 13% and 31% faster than CBT. PIM-SM has about the same delay characteristics as CBT, although, had processing delay been considered, PIM-SM would have lagged behind CBT.

- Join time: There was very little difference between any of the algorithms.

- Bandwidth efficiency: For both PIM-SM with shared trees and PIM-SM with SPT's only about 4% of the total bandwidth was needed to deliver data. PIM-DM is very bandwidth inefficient, but this improves greatly as the network becomes densely populated with group receivers.

- Overhead comparisons: CBT exhibited the lowest overhead percentage, even less than PIM-SM with shared trees. PIM-DM was shown to have more than double the overhead of PIM-SM with SPT's due to the periodic broadcasting and pruning. Table 3.1 gives the estimated number of routing table entries required at each router for a particular network environment. N is the number of active multicast groups in this network environment; n is the average number of senders to a group. Notice the significantly lower number of entries required by CBT for all scenarios.

Protocol	Group Size Parameters			
	N = 1000 n = 10	N = 1000 n = 200	N = 20,000 n = 10	N = 20,000 n = 200
CBT	500	500	10,000	10,000
PIM-DM	10,000	200,000	200,000	4,000,000
PIM-SM w/SPT	8,000	150,500	160,000	3,010,000
PIM-SM, shared tree	1,000	1,500	20,000	210,000

Table 3.1. Number of Router Table Entries Required by Protocols. After (Ballardie, 1996).

- Complexity comparisons: Based on subjective comparisons, CBT was judged to be the least complex of the protocols.

The Harris simulation also confirmed that traffic becomes concentrated on links near the set of core routers, if the number of cores is small. Increasing the number of cores can help alleviate this problem. (Ballardie, 1996)

c.   *Wei, Estrin Study*

The simulations of interest from the study were run on random networks of 50 and 200 nodes. For each network that was generated, a randomly selected multicast group was created, and shortest path and center-based trees were built for the group. Five hundred runs were made for each scenario. (Wei, 1994)

The three metrics used to evaluate the different approaches were end-to-end delay, cost, and traffic concentration. Cost included bandwidth consumed and router state. As in the Harris study, the authors expected that network size, multicast group size, and the number of sources sending to a group could affect the performance of each delivery algorithm. They also expected that the proportion of short links to long links (reasonableness of the graph), average number of nodes connected to each node (node degree), and the distribution of senders and receivers could influence the results. (Wei, 1994)

In the simulations run to collect delay and cost data, the parameters varied were network size, multicast group size, reasonableness of the graph, and node degree. In all cases, the delay of center-based trees was higher than shortest path trees and the cost associated with center-based trees was less than shortest path trees. Figure 3.6 shows the effect of group size on delay and cost; Figures 3.7 and 3.8 show how varying node degree impacts average delay and cost, respectively. In the Figures, a ratio above 1 indicates that CBT had higher values than SPT; the opposite is true for ratios less than 1.

Effect of Group Size in 50-node Network

Ratio (CBT/SPT)

1.4
1.2
1
0.8
0.6

0   10   20   30

Group Size

—◆— Maximum Delay
■ Average Delay
—▲— Cost

Effect of Node Degree in 50-node Network

Ratio (CBT/SPT)

1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8

0 2 4 6 8

Node Degree

—◆— Average Delay

Effect of Node Degree in 50-node Network

Ratio (CBT/SPT)

1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5

0 2 4 6 8 10

Node Degree

—◆— Cost
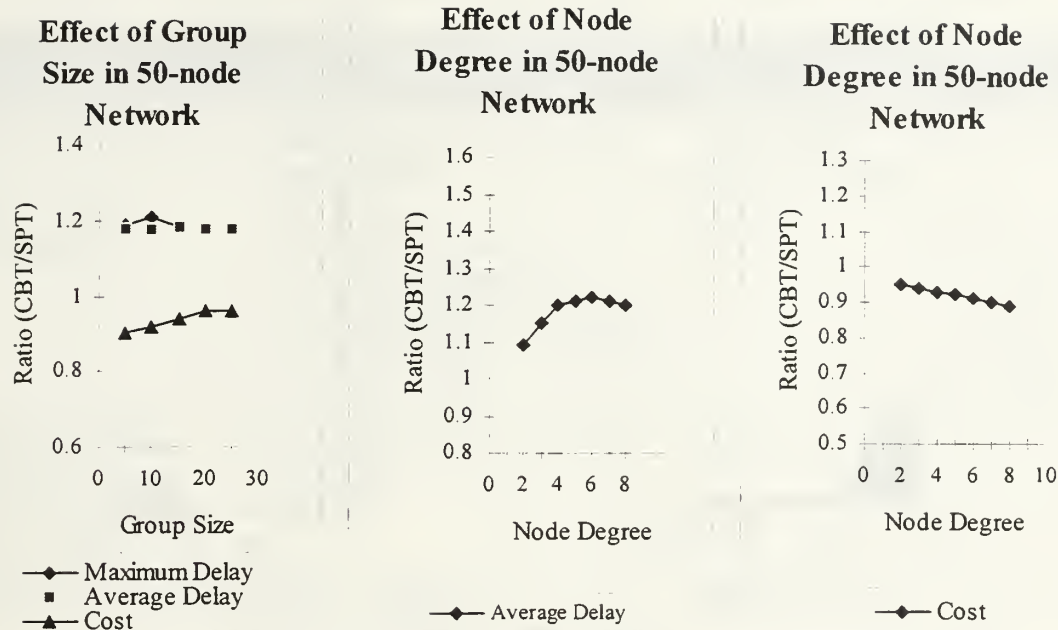
Figure 3.6. After (Wei, 1994).   Figure 3.7. After (Wei, 1994).   Figure 3.8. After (Wei, 1994).

For the traffic concentration experiments, node degree, number of senders, and distribution of members was varied. In networks with low connectivity, the two distribution algorithms had similar link utilizations. "However, when the average node degree increases, center-based trees maintain almost flat maximum link loads, whereas the maximum link load of shortest path trees drop significantly." (Wei, 1994) The authors concluded that link utilization dropped for shortest path trees as the connectivity increased because they made better use of alternative paths than center-based trees. The basic shape of each tree's link load histogram did not change as the number of senders was varied from 2 to 20. The SPT profile (Figure 3.9) shows more links being lightly loaded and a smaller number of links with high loads; the CBT histogram (Figure 3.10) suggests that some links are underutilized while others are overloaded (Wei, 1994).

SPT Link Load
Distribution

CBT Link Load
Distribution
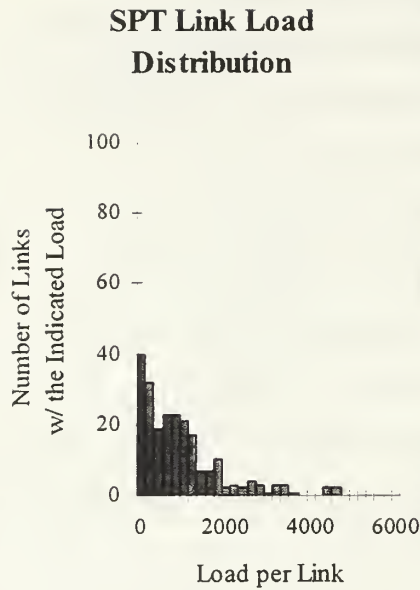
Figure 3.9. After (Wei, 1994).
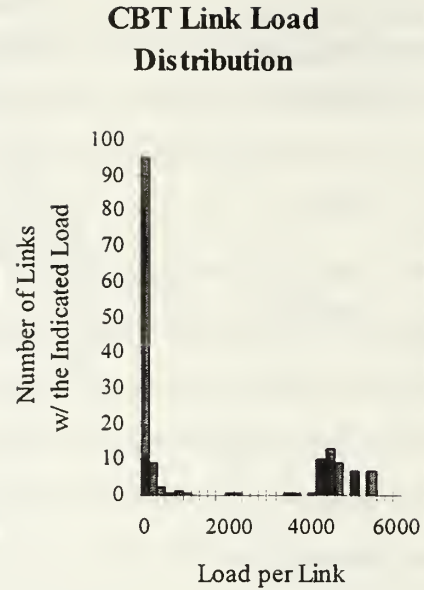
Figure 3.10. After (Wei, 1994).

Although the "…link load distribution is closely related to the distribution of member locations…", they did not vary much with changes in member distribution. (Wei, 1994) The traffic concentration results may appear to indict center-specific trees; however, the shared trees set up during the simulations had only one center. Shared trees with additional centers might have produced less damning results.

E. CONCLUSIONS

The protocol evaluations discussed throughout this chapter are summarized in Table 3.2.

Evaluation Criteria	SOURCE-SPECIFIC TREES	CENTER-SPECIFIC TREES
Router State	High; may be stored in "off tree" routers	Low: CBT High: PIM-SM
How State is Maintained	Soft (tree reflects changes in topology, but changes are propagated slowly (DVMRP), require complex calculations (MOSPF))	Hard
Control Traffic Overhead	High; may involve "off tree" routers	Low
Data Distribution Overhead	Higher	Low
Join Latency	High: MOSPF Same: PIM-DM, PIM-SM w/ SPT	Same
End-to-end Delay	Low	Higher, but bounded PIM-SM higher than CBT
Traffic Characteristics	Better distribution, especially in highly connected network	May become concentrated on links near core routers
Protocol Independence	No: DVMRP, MOSPF Yes: PIM-DM, PIM-SM w/ SPT	Yes
Central Point of Failure	No	Yes, but can be mitigated with multiple cores
Complexity	MOSPF complex PIM-SM most complex	CBT least complex, PIM-SM most complex
Other		CBT: hosts with non-multicast capable routers can send to group CBT, PIM-SM: no mechanism yet for core placement

Table 3.2. Evaluation of Routing Protocols Which Build Source-Specific and Center-Specific Trees.

Although the expected high overhead of protocols that build source-specific trees seem to make them unsuitable for the tactical internet, some applications may require the low end-to-end delay that they provide. Their tendency to better distribute traffic among links, and their lack of a central point of failure are attributes important in the threat-rich tactical environment.

Protocols that build center-specific trees, particularly CBT, appear more apt for the tactical internet based primarily on their low control traffic overhead. Nevertheless, some applications may be intolerant of the delays introduced by these protocols. The concentration of traffic near core routers may also make these protocols less robust than those that build source-specific trees.

In the final analysis, which protocol is selected depends on how the decision maker weights the requirements imposed by the tactical internet. If minimizing bandwidth is most important, the choices are narrowed to those protocols which construct center specific trees; if robustness is the most heavily weighted criteria, protocols in the other half of Table 3.2 may be more appropriate.

Similarly, the types of applications hosted on the tactical internet will influence the multicast routing protocol chosen. If a tactical data system will involve large numbers of senders and receivers, the protocol chosen must be scalable. How tolerant applications are of end-to-end delay will also determine what type of multicast routing protocol is selected for TDN.

IV. RELIABLE MULTICAST ISSUES

A.   INTRODUCTION

The multicast routing protocols of the previous chapter set up and maintain distribution trees for multicast IP traffic. Some applications, however, demand more reliability than the "best effort" delivery provided by multicast IP. The transport layer multicast protocols which have been developed to meet this demand are the subject of the next two chapters.

Since the definition of reliability drives protocol design, the first section discusses different perspectives on this subject. Next, arguments for moving some functions to the upper layers of the protocol stack are presented to help the reader understand the tradeoffs of guaranteeing reliability at the transport layer. The major emphasis of this chapter, though, is on the design choices which have been made by those who have developed reliable multicast protocols. Whenever possible, the technical impact of these choices is also discussed. The taxonomy which emerges will be used to classify the protocols of the next chapter.

B.   DEFINITIONS OF RELIABILITY

To Garcia-Molina and Spauster (1991), reliability is determined by three properties of a protocol. The first is the length of time it takes to deliver a multicast message, either measured from message initiation at the source or from the time the first message arrives at a group member. Delivery time may be guaranteed at one group member, a majority of group members, or all group members. The second property, atomicity, requires that all group members deliver a message to the application within a specified interval begun either after one group member has delivered the same message or after a majority of group members have delivered the message. The final property is ordering, which may range from no guarantees of ordering, to a guarantee that messages are always delivered consistently even in the presence of network failures.

Rajagopalan's (1992) definition of reliability is not as comprehensive or stringent as the definition used by Garcia-Molina and Spauster. Protocols are reliable if they offer

completeness and finiteness. Completeness means that the protocol delivers multicast messages "…in the same order as sent by the source, without message duplication or loss." (Rajagopalan, 1992) No mention is made of ordering guarantees in the presence of failures. Finiteness is similar to atomicity.

Bormann (1994) defines reliability from both the sender's and receiver's perspectives. A protocol is reliable if a receiver can determine when "…it is failing or being partitioned from active senders." (Bormann, 1994) From the sender's perspective, a protocol is reliable if the sender "…is assured (with sufficient probability) that all its messages reach within bounded time all recipients that are not failing or being partitioned." (Bormann, 1994)

(Macker, 1996) introduces a state dependent reliability which "…is a common requirement for many data types in advanced distributed simulations and in situational awareness dissemination." In these types of applications, data may be in either an active or quiescent state. For a data object which is changing relatively quickly (active state), reliable data delivery is not necessary since lost data will be refreshed with the next update. When a data object enters a steady state (or becomes quiescent), its state is periodically advertised to the multicast group. For more conventional reliable protocols, error recovery is initiated when a lost or misordered packet is detected; for the type of reliability described by Macker, action is taken only when an inconsistency is detected between subsequent packets. When this happens, "…appropriate action is taken to reliably establish a common 'distributed state'." (Macker, 8/9/96 e-mail)

Because the Garcia-Molina and Spauster definition of reliability is the most comprehensive, aspects of it will be used in the taxonomy developed later in this chapter. According to their definition, a protocol registers somewhere on the continuum for each property; the extent of a protocol's reliability is determined by aggregating its "rating" for each property. For example, protocols which are completely unreliable would make no guarantees of delivery time, atomicity, or ordering; completely reliable protocols, on the other hand, would guarantee delivery times to all group members, atomicity, and consistency even in the presence of failures. Protocols with less than complete reliability would fall somewhere between these two extremes.

The reliability that an application demands from a protocol determines many aspects of a protocol's design and subsequent performance, including its complexity, efficiency, overhead, and the amount of state which it requires senders and receivers to maintain. The type or level of reliability guaranteed by a protocol also determines how well suited it may be for certain types of applications.

C. WHERE RELIABILITY FUNCTIONS BELONG IN THE PROTOCOL STACK

1. The End-to-End Argument

The end-to-end argument "...appeals to application requirements and provides a rationale for moving a function upward in a layered system closer to the application that uses the function." (Saltzer, 1984)

> The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the endpoints of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.) (Saltzer, 1984)

When a file is transferred, for example, it is read from disk and passed to the file transfer program before being delivered to the communication subsystem. If the file is altered either during the read operation, or during its manipulation by the file transfer program, even a perfectly reliable communication subsystem will deliver a file different from the original. Only an end-to-end check, at a level higher than the delivery system, will be able to guarantee file integrity. Some communications reliability is necessary, though, to transfer the end-to-end checks. How much reliability to build into the communication subsystem is "... an engineering trade-off based on performance, rather than a requirement for correctness." (Saltzer, 1984) Too much reliability may increase protocol overhead and add delays; too little reliability may have a similar effect because of more frequent retransmissions. (Saltzer, 1984)

How much reliability to provide at the different levels of the protocol stack should be decided by the application. Applications which already check for errors, such as banking programs, may not need the communication protocol to provide perfect

reliability. Other applications which are more tolerant of the delays introduced by error checking at the lower levels might benefit from a highly fault tolerant delivery system.

2. Application Level Framing (ALF)

Application level framing is another argument for moving some functions higher in the protocol stack. The first three sub-sections provide background information intended to explain why ALF focuses on uninterrupted presentation conversion. The last subsection is devoted to application level framing.

a. *Protocol Functions*

"The core function of protocols is to transfer application information among machines." (Clark, 1990) Those functions performed during the transfer of this application information can be categorized as either data manipulation or transfer control. Data manipulation functions, which read or modify data, include: moving data to and from the network, detecting errors, buffering data for retransmission, encryption, moving data to and from application address space, and presentation formatting. Presentation formatting can be further defined as the reformatting of data into "...some common or external data representation." (Clark, 1990) Presentation conversion, which will be discussed later, includes both presentation formatting and moving data to and from application address space.

Transfer control operations are directly related to the transfer of data. They include: flow/congestion control, detecting network transmission problems, sending and processing acknowledgments, multiplexing, timestamping packets, and framing data. "In-band" controls are tightly linked to the actual data transfer; "out-of-band" controls do not have to be performed concurrent with transfers. (Clark, 1990)

b. *Performance Impact of Protocol Functions*

According to Clark and Tennenhouse (1990), for most protocols "...not many instructions are required for the in-band control operations..." and few of the control steps are computationally complex. "In contrast, the data manipulations are much more processing intensive, since they involve touching all the bytes in the packet, perhaps several times." (Clark, 1990) The higher cost of manipulating the data compared to the

44

cost of controlling its transfer makes data manipulation the obvious focus of efforts to enhance protocol performance.

c.   *Presentation Conversion*

The data manipulation step which most impacts performance is presentation conversion. In one experiment involving Unix TCP, 97% of the protocol overhead was due to presentation conversion. In another experiment conducted by the authors, various types of presentation conversions ran 4 - 5 times slower than an operation designed to minimize manipulations. (Clark, 1990)

"A key aspect of presentation conversion is that it needs to be done in the context of the application." (Clark, 1990) For example, when a file is transferred, the received data is simply placed in sequential locations in the application address space. Other scenarios are more complicated, however. "In some cases, only the application will know what the sequence of data items is, so that the actual sequence of presentation conversions must be driven by application knowledge." (Clark, 1990) This integral relationship between presentation conversion and the application makes the application a potential bottleneck in overall network throughput. Given this relationship, Clark and Tennenhouse conclude that protocols should be designed so that the "...application is not prevented from performing presentation processing as the data arrives." (Clark, 1990)

Presentation conversion is interrupted by lost or misordered data in most conventional protocols since incoming packets are not delivered to higher layers in the protocol stack until they have been properly sequenced by the protocol. Achieving the proper sequence may be a comparatively lengthy task since, in the case of lost packets, the sender is required to retransmit data. "Thus, a lost packet stops the application from performing presentation conversion, and to the extent it is the bottleneck, it can never catch up." (Clark, 1990)

d.   *Application Level Framing*

Application level framing shifts the design focus for protocols from transmission units that are understood by lower layers in the protocol stack to units which are meaningful to the application. Protocols designed according to ALF principles break the data into Application Data Units (ADU's) which can be processed by the application

even if they are out of order. Because presentation processing can continue uninterrupted despite out of sequence ADU's, protocol performance is boosted. (Clark, 1990)

For the receiving application to adjust to out of order ADU's, it must know where to put the arriving data. This information is communicated by the sender who "...must be able to specify the disposition of the ADU in terms meaningful to the receiver." (Clark, 1990) For example, in the case of a file transfer, the sender would transmit both the ADU and the ADU's location within the receiver's file. Out of sequence ADU's would not have to be reordered at the transport layer before being delivered to the application since the application would be able to determine their proper place within the receiver's file. Similarly, for video data the sender and receiver might agree on a code which identifies both the frame to which an ADU belongs and its location within that frame. (Clark, 1990)

Since the size of the ADU is application specific, Clark and Tennenhouse only provide guidelines for determining its length. However, both Crowcroft (1996) and Heybey (1991) have developed protocols based upon ALF. For Crowcroft's text editor (nt), a line of text is the ADU; Heybey develops ADU's for each of the video coding algorithms he studied.

The opinion expressed in (Floyd, 1995) is that, for multicast communications, the ADU is a more meaningful descriptor of data than the numbered packets typically used in conventional unicast protocols such as TCP. Because a unicast session has a starting point known by both the sender and receiver, sequence numbers describe the progress of the session. Sequence numbers are more ambiguous in multicast communications since receivers are allowed to join in-progress sessions. After joining a session late, a receiver may be unable to determine how far the conversation has progressed from the numbered packets it receives. Without knowing this, the receiver will be unable to process subsequent packets or request retransmissions for the packets it has missed. (Floyd, 1995)

ALF seems to have evolved from a narrow concept concerned primarily with how data is aggregated, to a much broader interpretation. Floyd (1995) writes: "ALF says that the best way to meet diverse application requirements is to leave as much

functionality and flexibility as possible to the application." Macker (5/22/96 e-mail) also suggests that ALF is more encompassing than originally proposed since according to ALF principles "…some reliable multicast design details belong at the application layer…."

The importance of this discussion of application level framing is summed up by Macker's next statement regarding reliable multicast transport protocols: "There is no one magic transport layer solution." (Macker, 5/22/96 e-mail) Some aspects of reliability are best handled at the transport layer; others belong higher in the protocol stack. Furthermore, changing the transmission unit paradigm from packets understood by the transport layer, to ADU's, can lead to gains in protocol performance.

D. TAXONOMY OF RELIABLE MULTICAST DESIGN CHOICES

The taxonomy developed in this section will be used to classify the reliable multicast protocols in the next chapter. The design choices mentioned here are features of existing multicast protocols.

1. Error Recovery

a. *Introduction*

The methods of guaranteeing reliable data delivery are distinguished by who is responsible for detecting errors, how errors are signaled, and how missing data is retransmitted (MIST website). Error recovery schemes are divided into two broad classes based upon the party responsible for detecting errors. Characteristics of sender-initiated protocols are discussed first, followed by general comments about receiver-initiated protocols. The remainder of this section is spent explaining variants of the receiver-initiated protocol.

b. *Sender-Initiated (or Sender-Reliable) Error Recovery*

Sender-initiated error recovery is an extension of the unicast reliability mechanism. Receivers acknowledge each message; ACK's are unicast to the sender who maintains state for all receivers in the multicast group. Timers are managed at the sender to detect packet losses. Only missing packets are retransmitted, either to individual receivers, or to all receivers.

The weaknesses of this approach limit its use. Senders may become overwhelmed by the simultaneous receipt of ACK's, or by the requirement to process

ACK's from a large number of receivers ("implosion" effect). The amount of state which the sender must maintain to track the receiver set may also grow prohibitively large. "Finally, the algorithms that are used to adapt to changing network conditions tend to lose their meaning..." since timing and control parameters must be calculated for a potentially diverse set of receivers. (Floyd, 1995)

Sender-initiated error recovery is limited to small multicast groups because of the ACK implosion effect and the requirement that the sender maintain state for all receivers. However, (Macker, 1996) feels that this approach may be appropriate in situations where the sender must be in absolute control.

c. *Receiver-Initiated (or Receiver-Reliable) Error Recovery*

The responsibility for reliable delivery is shifted from the sender to receivers in receiver-initiated error recovery. "Each receiver maintains reception state and requests repairs via a negative acknowledgment (NACK) when an error is detected." (Macker, 1996) Low latency detection of losses requires frequent transmissions; otherwise, some type of periodic "heartbeat" transmission is necessary (Macker, 1996).

Receiver-initiated error recovery reduces the processing burden of the sender, and frees the sender from the responsibility of maintaining state for all receivers. Another advantage of this approach is that the receiver decides what level of service to provide to the application. This arrangement is a natural fit since "...the receiver best knows the type and quality of services desired...." (Pingali, 1994)

d. *Variations of Receiver-Initiated (or Receiver-Reliable) Error Recovery*

The variants to receiver-initiated error recovery will be described next.

(1) Sender-Oriented. The two types of sender-oriented error recovery methods are distinguished by how they transmit NACK's to the sender. The unicast version will be described before the broadcast variant.

(a) Unicast NACK. When receivers detect an error, a NACK is unicast to the sender. Only the sender is involved in issuing repairs. (Macker, 1996) In the sender-oriented protocol of (Pingali, 1994), retransmitted packets are given priority over new packets, and are multicast to the group.

(Pingali, 1994) used simulations to compare the maximum throughput of sender-initiated error recovery to receiver-initiated approaches. Figure 4.1 plots the throughput at the sender for their receiver-initiated/sender-oriented protocol (N1) divided by the throughput at the sender for their sender-initiated protocol (A). Values greater than 1 indicate higher maximum throughput at the sender for the receiver-initiated protocol. Each plot represents a different probability of packet loss; loss probabilities of 1, 5, 10, 25, and 50 percent were simulated.

Comparison of Sender Throughputs

Figure 4.1. Comparison of Sender Throughputs for a Receiver-Initiated/ Sender-Oriented Protocol and a Sender-Initiated Protocol. After (Pingali, 1994).

"For a given loss probability, the relative performance of..." the receiver-initiated protocol "...improves over that of..." the sender-initiated protocol as the number of receivers is increased. (Pingali, 1994) The difference in maximum sender throughput of the two error recovery strategies can be explained by the amount of processing each requires at the sender. A receiver-initiated/sender-oriented approach only burdens the sender when losses occur. For sender-initiated error recovery, acknowledgments must be processed at the sender even if communication is error free; the amount of processing increases with an increase in the number of receivers. (Pingali, 1994)

Figure 4.2 plots the maximum throughput at the receivers for the same two protocols. For loss rates of up to 50%, the receiver-initiated protocol outperforms the sender-initiated protocol. Throughput is higher for the receiver-initiated approach because receivers only issue NACK's when losses occur; the sender-initiated protocol requires receivers to acknowledge each packet.

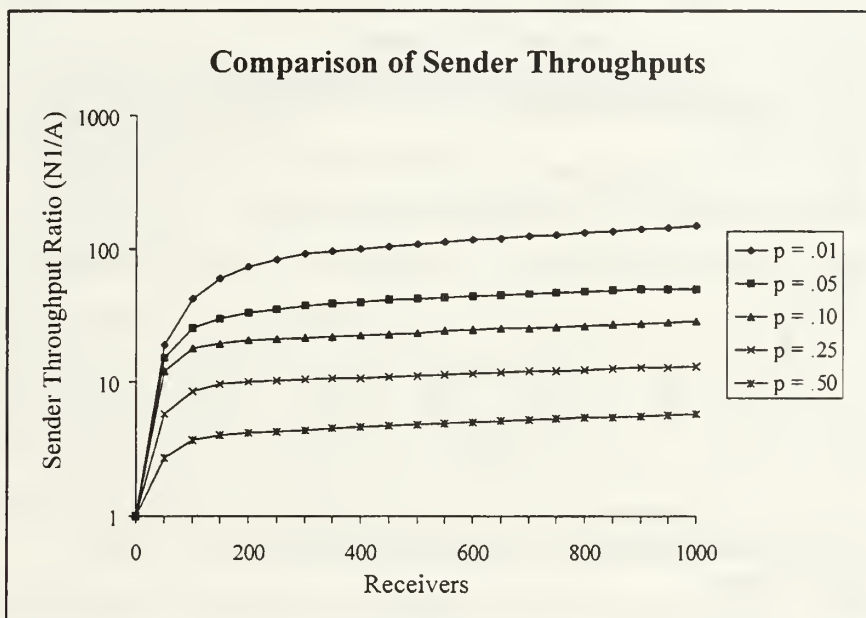**Comparison of Receiver Throughputs**
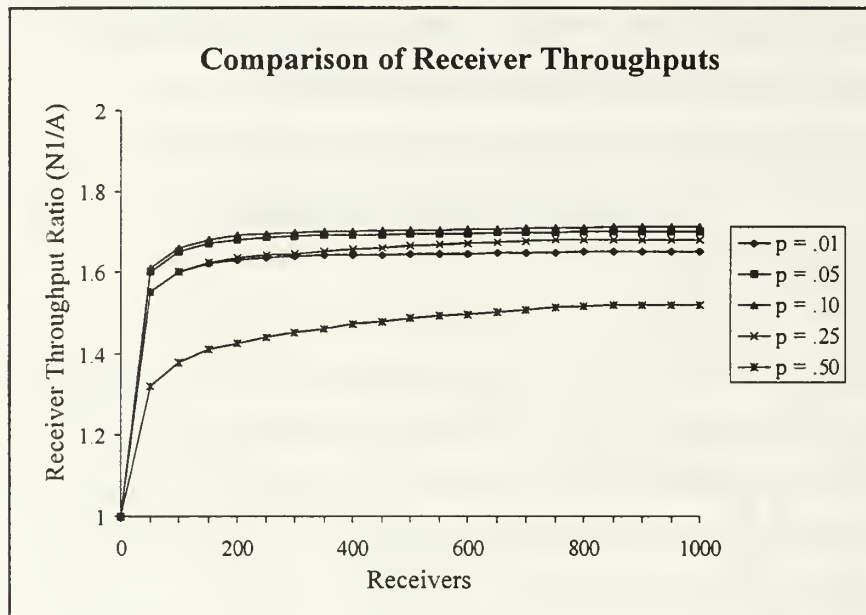
Figure 4.2. Comparison of Receiver Throughputs for a Receiver-Initiated/ Sender-Oriented Protocol and a Sender-Initiated Protocol. After (Pingali, 1994).

(Macker, 1996) simulated packet loss in a network to study the effects of NACK implosion for receiver-initiated/sender-oriented error recovery. Figure 4.3 is a plot of the results. The parameter d represents the depth of the multicast tree; n represents the fan-out. Since n was held constant at 4, higher values of d indicate more multicast group members. As Figure 4.3 shows, the number of NACK's increases with an increase in both the number of multicast members and the probability of packet loss. The increased number of NACK's will have two effects: NACK implosion will eventually overwhelm the sender, and more bandwidth, over all links of the network, will be consumed by control traffic and retransmissions.

NACK's Generated by Simulated Receiver-Initiated/Sender-Oriented Protocol
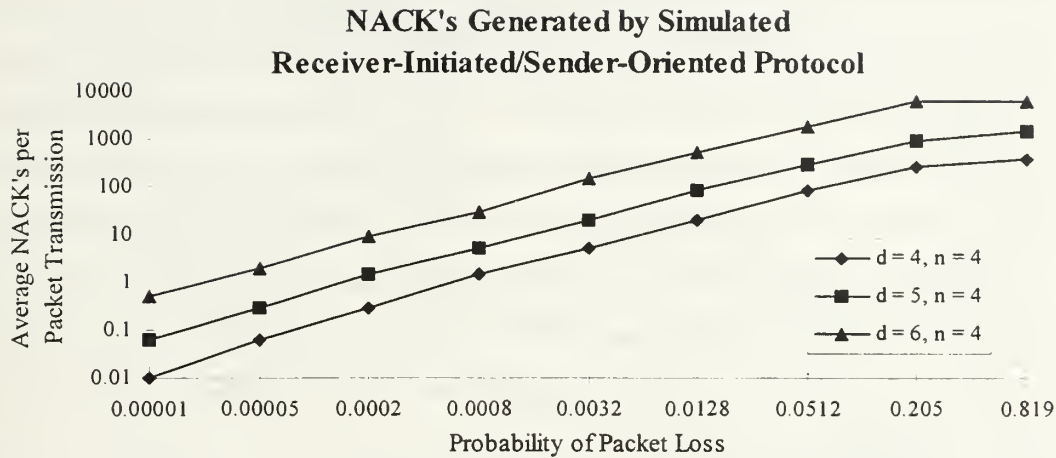
Figure 4.3. NACK's Generated by Simulated Receiver-Initiated/Sender-Oriented Protocol. After (Macker, 1996).

Receiver-initiated/sender-oriented error recovery may be appropriate when receivers are not permitted to communicate with each other (Macker, 1996). This approach is more scalable than sender-initiated error recovery because it reduces the processing load at the sender. However, the results of the simulation conducted in (Macker, 1996) suggest that this approach may not scale to large receiver sets.

(b) Broadcast NACK. This variant "attempts to ensure that at most one NAK (sic) is returned to the sender per packet transmission." (Pingali, 1994) When a receiver detects an error, it waits a random period of time before broadcasting a NACK to the sender and all other receivers. A receiver will cancel its broadcast if it receives a NACK which corresponds to a packet it has missed. Repairs are multicast to the group by the sender. (Pingali, 1994)

A comparison of the two sender-oriented error recovery strategies can be found in Figure 4.4. The plot is of the maximum throughput at the sender for the broadcast version of the sender-oriented protocol (N2) divided by the maximum throughput at the sender for the unicast version (N1). Values greater than 1 indicate higher maximum throughput at the sender for the broadcast variant. The simulation was run for the same loss probabilities as before.

51
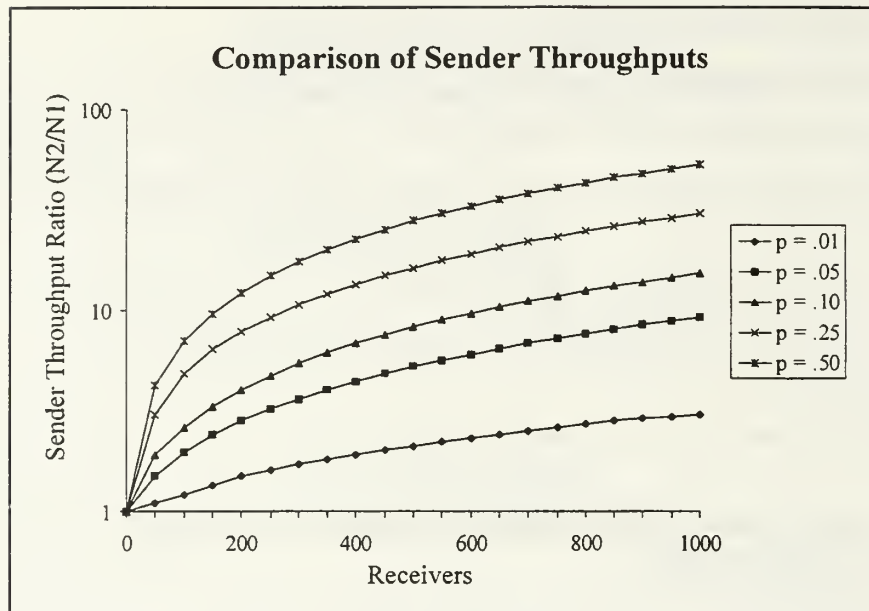
Figure 4.4. Comparison of Sender Throughputs for a Receiver-Initiated/ Sender-Oriented (Broadcast NACK) Protocol and a Receiver-Initiated/ Sender-Oriented (Unicast NACK) Protocol. After (Pingali, 1994).

The graph confirms the expected similar performance of the two recovery strategies for low numbers of receivers and low loss probabilities. As both the number of receivers and the loss probability increases, the broadcast version outperforms its unicast competitor. With the broadcast approach, the sender is only burdened with processing one NACK for each lost packet, instead of having to process a NACK from each affected receiver. Although the broadcast variant will consume more bandwidth in a low loss environment, it will support more receivers because of the processing it eliminates at the sender.

(2) Flat Receiver-Oriented. "In a flat receiver-oriented approach to reliable multicast, receivers can communicate with each other to assist in error recovery." (Macker, 1996) Much like the sender-oriented (broadcast NACK) approach, upon detecting an error, a receiver waits a random period of time before multicasting a NACK to the group. This random delay prevents receivers equidistant from the source from issuing redundant NACK's (Macker, 1996). Other receivers affected by the lost packet suppress their NACK if they receive a NACK corresponding to the same missed packet.

Since receivers cache data, they can also issue repairs. Receivers

52

that have cached missing data can randomly delay their repair to prevent redundant repairs from being issued by equidistant receivers whose NACK's arrive simultaneously (random suppression) (Macker, 1996). In a linear topology where downstream receivers detect the same errors as upstream receivers, the timers of downstream receivers can be adjusted to prevent redundancy (deterministic suppression) (Macker, 1996). "Since most networks exhibit both linear and star (equidistant) characteristics, a combination of randomized and deterministic NACK/repair suppression should be used for a flat, receiver-oriented reliability scheme." (Macker, 1996)

The flat receiver-oriented approach consumes similar amounts of bandwidth as the sender-oriented (broadcast NAK) approach since NACK's and repairs are global in scope. "Enhanced localized scoping of repair messages is possible and can alleviate this effect." (Macker, 1996) This approach would also seem able to support a similar number of receivers as its sender-oriented (Broadcast NACK) sibling, but would be more fault tolerant since each receiver is capable of issuing repairs. The cost of this increased fault tolerance, however, is that receivers are required to cache data.

(3) Hierarchical Receiver-Oriented. The hierarchical and flat receiver-oriented approaches are similar in that receivers communicate with each other to assist in error recovery. The difference between the two methods is that only designated receivers assist in error recovery in the hierarchical approach. As the name suggests, these designated receivers are organized into an error recovery hierarchy within the multicast delivery tree.

Several hierarchical error recovery algorithms have been developed. The method adopted by the Tree-based Multicast Transport Protocol (TMTP) will be discussed in the next chapter. Macker and Corson have developed an approach which is similar to TMTP; another approach "…forms a hierarchy of caching loggers, to which a receiver NACK's for a repair." (Macker, 1996) The error recovery method discussed in (Lucas, 1995) was developed for time-sensitive audio and video applications operating over wide area packet-switched networks. Although it does not provide fully reliable multicast, their approach will be described in greater detail here because it addresses issues important in a tactical internet.

In the (Lucas, 1995) protocol, designated receivers, called retransmission agents, are associated with a set of passive receivers. When a retransmission agent detects a lost packet, it sends a retransmission request to one or more other agents. An agent which has cached a copy of the packet forwards it to the requesting retransmission agent. The retransmission agent then multicasts a copy of the recovered packet to its passive receivers. (Lucas, 1995)

"The effectiveness of distributed retransmission depends on developing a mechanism to group together multicast receivers with similar error characteristics." (Lucas, 1995) The characteristics of commercial packet-switched WAN's act as this mechanism, resulting in groups of receivers with shared reception characteristics (Lucas, 1995). The same can be said of the tactical internet where high bandwidth, error free LAN's are connected to point-to-point wide area links with lower bandwidth and greater susceptibility to disruption. Consequently, losses from congestion and jamming are more likely in the wide area links than in the attached end systems. These losses affect all receivers on networks which are attached to the wide area links after the point of loss.

Since retransmission agents act on behalf of receivers with similar error characteristics, at least one agent should be located on each local area network. An election algorithm may select retransmission agents or they may be chosen manually. At regular intervals, each retransmission agent notifies other agents of its location and availability for answering retransmission requests. The bandwidth consumed by these advertisements is 1 - 5% of the data stream. (Lucas, 1995)

The error recovery schemes examined in this section retransmit missing data. (Lucas, 1995) discusses two alternative methods for managing errors - rate control, and network layer forward error correction (FEC). Although rate control does not recover lost packets, it does prevent congestion based losses. Forward error correction adds redundant information which allows lost packets to be reconstructed from correctly received packets. The advantage of FEC is that packets can be recovered without the delay associated with retransmissions; the primary disadvantage is that the redundancy consumes extra bandwidth.

The simulation topology built to test the performance of this protocol resembled the regional network which connects 17 academic and commercial campuses across the southeastern United States (Lucas, 1995). Application traffic was a video stream averaging 220 Kbits/sec, with peak rates up to 530 Kbits/sec. Comparisons between the hierarchical error recovery approach and different levels of FEC were based upon a penalty value for each link which was calculated by multiplying the packet loss rate (congestion) of the link by the amount of protocol traffic on the link. The two factors are multiplied together to more severely penalize adding protocol overhead to congested links. (Lucas, 1995)

Figure 4.5 compares the cost of the hierarchical error recovery strategy (retransmission) to 10% and 21% overcoding in an isolated loss environment. Loss probability is represented by ρ. For an isolated loss environment, the link drop rates are between 0 - 2.7% for $\rho = 0$, and between 2.4 - 4.8% for $\rho = .024$ (Lucas, 1995).

Network Cost for an Isolated Loss Environment

Figure 4.5. Comparison of Network Cost for Different Error Recovery Mechanisms in an Isolated Loss Environment. After (Lucas, 1995).

Figure 4.6 makes the same comparison, but for bursty losses where the link drop rates range from 0 - 2.8% for $\rho = 0$ to 6.9 - 10.1% for $\rho = .024$ (Lucas, 1995). At low error rates, the penalty for hierarchical error recovery is low. "However, the network penalty increases rapidly as error rates rise, due largely to the increased amount of retransmitted data." (Lucas, 1995)

Network Cost for a Bursty Loss Environment

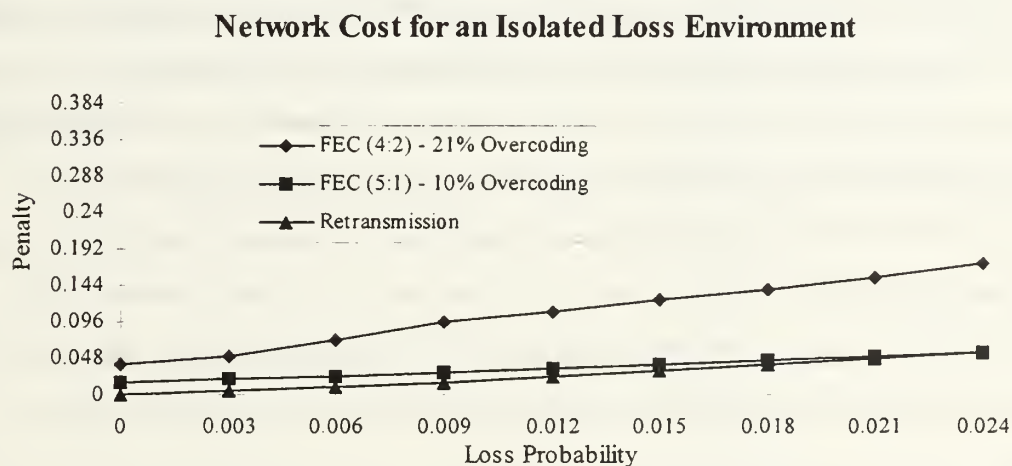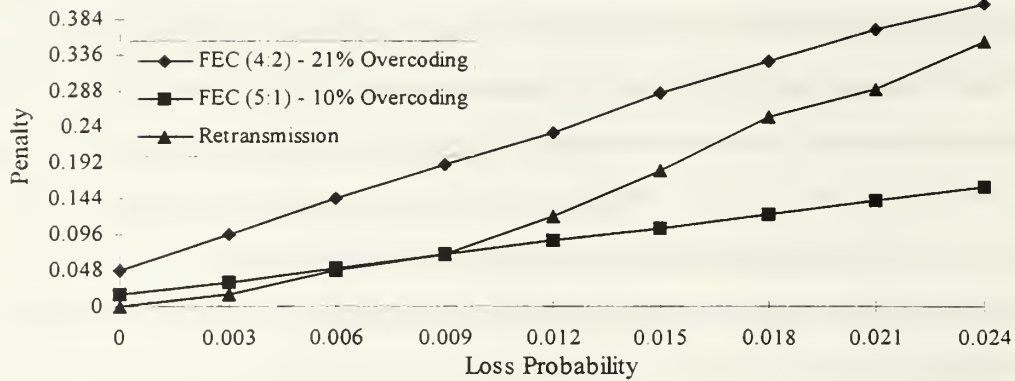Figure 4.6. Comparison of Network Cost for Different Error Recovery Mechanisms in a Bursty Loss Environment. After (Lucas, 1995).

The conclusions reached in (Lucas, 1995) also apply to other variations of hierarchical error recovery. Fewer receivers are burdened with caching data or assisting with error recovery than in the flat receiver-oriented approach. More importantly, requests for missing data and the corresponding repairs are confined to those areas of the network where the loss occurred. For time-sensitive applications, hierarchical error recovery reduces the latency of repairs compared to sender-initiated and receiver-initiated/sender-oriented approaches since missing packets are retransmitted from a local retransmission agent instead of a possibly remote source. The primary disadvantage of hierarchical error recovery "…is the complexity costs associated with setting up and managing…" the protocol. (Lucas, 1995) This approach may also be less resilient than the flat receiver-oriented scheme since fewer receivers assist with error recovery.

(4) Supporting Complete Reliability with Receiver-Initiated Error Recovery. A sender buffers data that has not been delivered reliably to all its receivers. Since senders do not know the state of their receivers in receiver-initiated error recovery, a sender supporting complete reliability may be required to buffer data indefinitely. An application with a bounded latency requirement will not impose this burden on the sender, since expired data can be discarded. In other cases, however, the sender may be unable to buffer data indefinitely, especially for long-lived sessions. (Macker, 1996)

56

To keep buffer size at a manageable level, receivers could periodically inform the sender of their state. Unneeded data could then be purged from the sender's buffers. This mechanism could be combined with receiver-initiated error recovery to reduce the processing burden at the sender. "Thus, any scheme supporting absolute reliability represents a mixed requirement of both sender-reliable and receiver-reliable multicast." (Macker, 1996)

d.   *Method of Retransmission*

For each type of error recovery strategy discussed in this section, repairs were multicast to the group. The other alternative is to unicast repairs. Unicasting a missing packet requires explicit knowledge of the receiver set by the retransmitter, whether it be the sender or a designated receiver in a hierarchical error recovery scheme. For low loss networks where failures are independent of each other, unicast retransmission may be appropriate (Koifman, 1996). Multicast retransmission impacts throughput less, but requires all receivers in the group to process the retransmitted data, whether they are missing it or not. For networks where receivers share the same loss characteristics, multicast retransmission may be more appropriate than unicast retransmission.

Tables 4.1 and 4.2 summarize the error recovery approaches discussed in this section.

Method of Error Recovery	Error Detection	Repair Scheme	SCALABILITY		
			Processing Burden	State Requirements	Protocol Overhead
Sender-Initiated	Receivers unicast ACK to Sender	Sender broadcasts (multicasts) repair	Sender: High for large receiver sets Receivers: Low	Sender: High for large receiver sets Receivers: Low	High: Global scope of repairs

Table 4.1. Characteristics of the Sender-Initiated Error Recovery Approach.

Method of Error Recovery	Error Detection	Repair Scheme	SCALABILITY		
			Processing Burden	State Requirements	Protocol Overhead
Sender-Oriented (Unicast NACK)	Receivers unicast NACK to Sender	Sender multicasts repair	Sender: Less than Sender-Initiated, but still high Receivers: Low	Sender: Less than Sender-Initiated Receivers: Low	High: Global scope of repairs
Sender-Oriented (Broadcast NACK)	Receivers broadcast NACK to Sender and all Receivers	Sender multicasts repair	Sender: Less than Sender-Oriented (Unicast NACK) - only one NACK per lost packet Receivers: More than Sender-Oriented (Unicast NACK)	Sender: Less than Sender-Oriented (Unicast NACK) Receivers: More than Sender-Oriented (Unicast NACK)	High: Global scope of NACK's, and repairs
Flat	Receivers multicast NACK	Receivers cache data and can issue repairs	Sender: Low Receivers: Higher	Sender: Low Receivers: Higher	High: Global scope of NACK's, and repairs
Hierarchical	Receivers NACK to a designated receiver	Designated receivers cache data, issue repairs	Sender: Low Receivers: Low Designated Receivers: Higher	Sender: Low Receivers: Low Designated Receivers: Higher	Low: Local scope of NACK's, and repairs

Table 4.2. Characteristics of Receiver-Initiated Error Recovery Approaches.

2. Heterogeneous Receivers

Multicast protocols "...must deal with the possibility that not all of the receivers have access to the same hardware and network resources." (MIST website) Receivers may have different processing capabilities, and may be reached by communication links of differing capacities. To adapt to this heterogeneous environment, a protocol should be able to offer different levels of reliability to receivers, and "...should have some fairness policy to dictate how the different receiver characteristics are balanced...." (MIST website)

3. Scalability

How many receivers a reliable multicast protocol can support is difficult to calculate. A qualitative estimate can be made by examining those aspects of the protocol which affect the amount of bandwidth consumed, the processing conducted at senders and receivers, and the amount of state which must be maintained. The error recovery strategy and the ordering guaranteed by a protocol seem to influence these factors most.

4. Flow Control

"Controlling the packet rate in multicasting is complicated by the fact that the protocol must accommodate multiple receivers simultaneously." (MIST website) The type of flow control policy implemented by a protocol is important because of the significant impact it can have on performance. (MIST website)

In a tactical internet, packets may be dropped because of poor quality links as often as they are dropped because of congestion. Although an increased number of retransmission requests may be an indication of either situation, different responses are required in each case. How a protocol interprets indicators typically used to adjust the transmission rate will determine if it is likely to respond appropriately in the tactical internet.

5. Late-join/Leave

Receivers may join an in-progress session, or leave the group before the session has been completed. How the multicast protocol adapts to these late-joins and early-departures may influence error recovery and flow control mechanisms (MIST website).

The amount of state maintained by the sender, the traffic load, and the amount of protocol overhead may also be affected.

6. Fragmentation/Reassembly

Some transport layer multicast protocols fragment and reassemble packets; others rely on higher layers to perform this task.

7. Ordering

a. Introduction

Although ordering is not required for reliable multicast, some level of reliability is necessary to guarantee ordering. The varying degrees of ordering may be provided by the application or by the communications protocol. If the application orders data, "…message timestamps and sequence numbers must be available at the application level." (Crowcroft, 1996) Ordering data before it reaches the application can "…simplify the design of distributed software and reduce the probability that subtle synchronization or concurrency related bugs will arise." (Garcia-Molina, 1991) Some reliable multicast protocols perform ordering at the transport layer; other researchers feel that ordering mechanisms belong between the application and transport layers (Mayer, 1992).

Ordering is included in the taxonomy of reliable multicast protocols for two reasons: some of the transport layer protocols include an ordering mechanism, and multicast applications may require ordering guarantees from the communications protocol. The categories of ordering discussed here are: single source ordering, causal ordering, and total ordering. Unordered delivery is omitted because of its obvious implications.

b. Single Source Ordering

Also called FIFO ordering (Kaashoek, 1992), or simply ordering (Macker, 1996), single source ordering ensures that data is delivered to the receiver's application in the same order as it was transmitted. Single source ordering requires that messages be assigned sequence numbers at the sender. Receivers guarantee the correct order by passing messages to the application in sequence. (Garcia-Molina, 1991)

c. Causal Ordering

"Causal ordering guarantees that all messages that are related are ordered." (Kaashoek, 1992) The relationship between messages is established by the application. Cooperative work, work flow management, and conferencing applications typically generate messages which are causally related (Aiello, 1993). For example, in a conferencing application, user A may transmit a message (m_1) to the other participants which solicits a response from user B (Figure 4.7). When user B responds with message m_2, the ordering mechanism will ensure that all participants have received message m_1 before m_2 (Figure 4.8). (Macker, 1996)
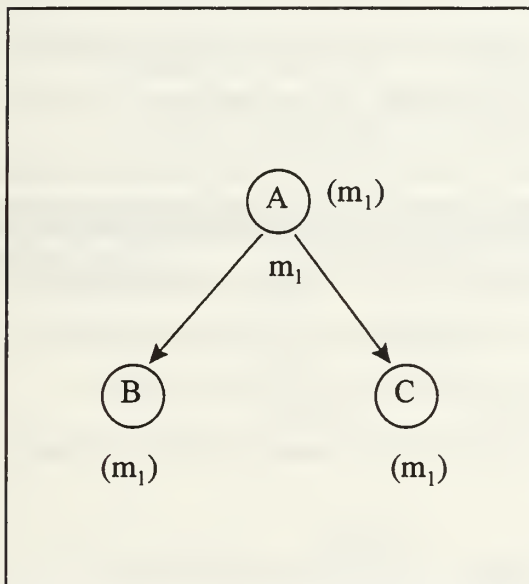
Figure 4.7. User A Multicasts Message m_1 to Receivers B and C.
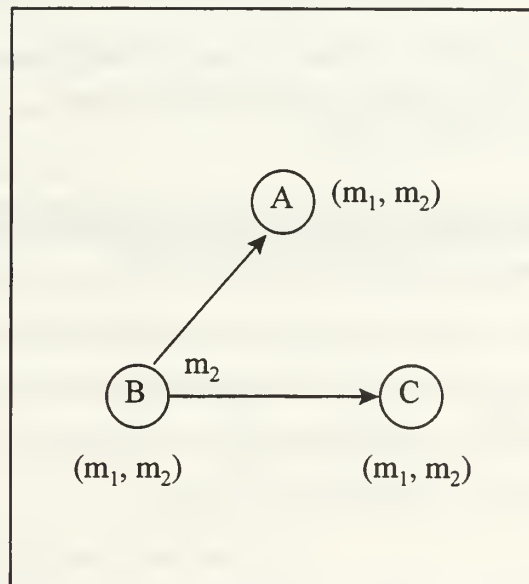
Figure 4.8. User B Multicasts Message m_2 to Receivers A and C.

d. Total Ordering

"Totally ordered delivery specifies that multiple multicast streams from multiple senders are delivered sequentially to each receiver and are received in the same relative order at each receiver." (Macker, 1996) This level of ordering is typically required in distributed systems. For example, if two users of a replicated database system make updates at different times to the same file stored on three different platforms (Figures 4.9, 4.10), total ordering ensures that each copy of the file reflects the relative order of the updates.
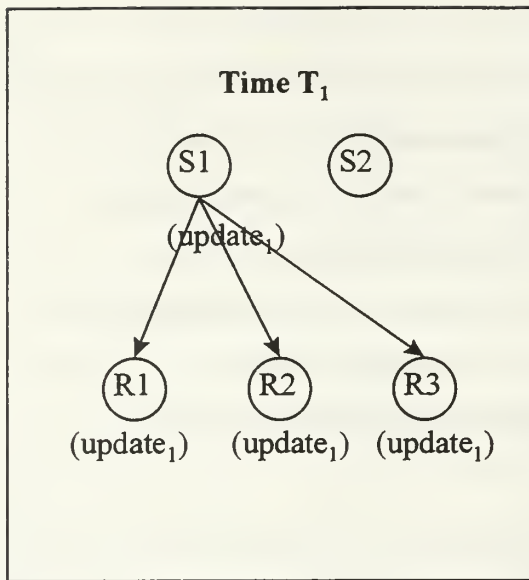
Figure 4.9. At Time T_1, Sender 1 Multicasts the First Database Update to the Group. After (Mayer, 1992).
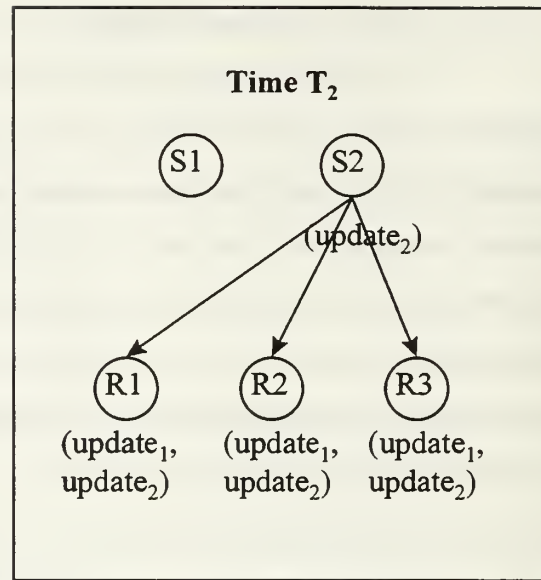
Figure 4.10. At Time T_2, Sender 2 Multicasts the Second Database Update to the Group. After (Mayer, 1992).

For totally ordered protocols, the ordering mechanism adds delay to that already provided by the network. Mayer (1992) gave the term synchronization delay to the "…excess delay that a protocol generates to achieve ordering." His study compared the synchronization delay of three types of totally ordered protocols for low traffic conditions. For high traffic environments such as the tactical internet, the delay values he calculated would be lower bounds.

(1) Sender-Controlled Protocol. The ordering mechanism of a sender-controlled protocol is similar to a two-phase commit. In the first phase, the sender multicasts a message to the receivers. After buffering the message and assigning it a priority, each receiver returns this priority to the sender. The second phase begins with the sender computing the maximum of these priorities and multicasting it back to the receivers. Based upon the priority scheme employed by a protocol, each receiver then delivers the same message to its application. (Mayer, 1992)

(2) Receiver-Controlled Protocol. Receiver-controlled protocols are more commonly known as token-based protocols. Senders multicast messages to all receivers in the group, including one designated as the primary receiver. The primary

receiver can deliver a message immediately; other receivers must wait until an "acknowledgment" message arrives from the primary receiver. (Mayer, 1992)

(3) Time-Controlled Protocol. The time-controlled approach "...is possible if network and communication software have realtime properties, in particular if there is an upper bound on the time for end-to-end message transfer." (Mayer, 1992) In these circumstances, the time-controlled protocol is able to totally order messages with very few synchronization messages (Mayer, 1992). Attached to each message multicast by the sender is a synchronization message containing a timestamp. From this timestamp and other information, each receiver calculates the latest time when the message must have arrived at all receivers. The message is delivered to the application at this time according to a local clock. (Mayer, 1992)

Mayer's conclusions about the best and worst environments for each totally ordered protocol are summarized in Table 4.3.
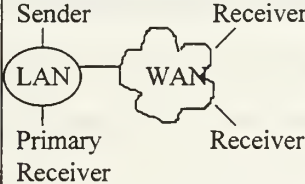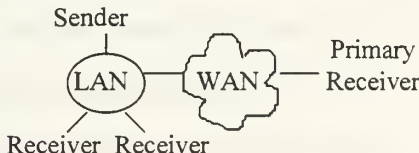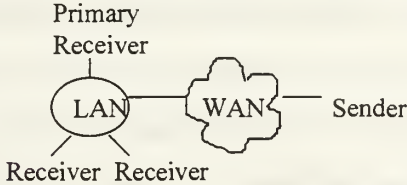
Protocol	Best Environment	Worst Environment
Sender-controlled	Small network delays: LAN	One slow network connection
Receiver-controlled	Sender Receiver / LAN — WAN \ Primary Receiver Receiver / Primary Receiver / LAN — WAN \ Sender Receiver Receiver	Sender / LAN — WAN \ Primary Receiver Receiver Receiver
Time-controlled	Homogeneous network delays: LAN's, specific types of WAN's	Varying network delays: very heterogeneous networks

Table 4.3. Best and Worst Environments for Different Types of Totally Ordered Protocols. After (Mayer, 1992).

Another important finding of Mayer's study is the effect that adding new receivers has on the synchronization delay of other receivers. Interestingly, if a receiver with a large network delay between itself and the sender joins a group, the synchronization delay of other receivers may increase for the sender and time-controlled approaches. The receiver-controlled approach is more resilient since the delay for a

receiver is dependent only on the connection between it and the primary receiver. The characteristics of new receivers joining the group have very little effect on the performance of other receivers. (Mayer, 1992)

8. Delivery Semantics

a. *Number of Deliveries to be Considered Successful*

One measure of the reliability of a protocol is when it considers a message to be successfully delivered to the multicast group. The three common choices are (Kaashoek, 1992):

- k-delivery - k processes receive the message.
- quorum delivery - the message is delivered to a majority of the receivers.
- atomicity - either all surviving processes receive the message, or none do.

b. *Time to Deliver a Message*

The reliability of protocols may also be defined, in part, by the length of time it takes to deliver messages (Garcia-Molina, 1991). Real-time protocols may guarantee delivery within a certain time from message initiation at the source. Other protocols which guarantee ordering may promise delivery to the application within a certain time after the message arrives at a group member. (Garcia-Molina, 1991)

9. Recovery from Failure

How a reliable multicast protocol behaves in the presence of failures is particularly important, especially for protocols operating in the threat-rich tactical environment. Protocols may be required to respond to failures in several different ways.

(Garcia-Molina, 1991) categorize the reliability of ordering guarantees in the presence of failures into four levels of increasing reliability:

- R0 - no guarantees of ordering.
- R1 - messages are delivered in a consistent order. Failed sites do not have to recover messages; order can be inconsistent at failed sites.
- R2 - after a failure, sites that have recovered must redeliver messages that were delivered inconsistently during the failure.
- R3 - inconsistent deliveries are never allowed.

After a failure, protocols may recover lost data or be required to reform the multicast group. Some applications may want to continue operating within a partition after receivers in a multicast group have been partitioned by a network failure (Bormann, 1994).

10. Prioritization of Traffic

The transport layer of some unicast protocols allows traffic to be prioritized. Reliable transport layer multicast protocols may be required to provide the same type of service to the transport user.

11. Group Structure

Multicast groups which allow non-members to communicate with the group are called open; those which exclude non-members are known as closed groups. If membership changes are permitted during a session, the group is dynamic; groups that do not allow receivers to join or leave during a session are static. (Kaashoek, 1992)

12. Concast or Unicast Capability

Some applications may require many-to-one, or concast, communications. For interactive collaborative applications, participants act as both senders and receivers. Sources multicast to the receivers in the group; group members may then return state information to these sources. Similarly, for sensor processing applications, sensors concast data to a central processor; the processor, in turn, multicasts control data to its sensors.

According to Buddenberg (7/31/96 e-mail) "…we need to treat multicasting as the norm and unicasting as simply a trivial instantiation of multicasting…." Whether a multicast protocol also provides unicast capability will be noted in this category.

13. Other

This catch-all category will be used to describe other notable characteristics of a multicast protocol which are not captured by the other categories.

E. EVALUATION CRITERIA

1. Design Choices

Reliable transport layer multicast protocols can be evaluated according to some of the same criteria which were used to evaluate the multicast routing protocols of the previous chapter:

- state requirements - costs to both senders and receivers.
- performance - possible metrics include: aggregate throughput, single sender throughput, end-to-end delay, processing load, and bandwidth consumed.
- efficiency
- technical scalability - how well a protocol accommodates evolutionary changes to its design

Drawing valid conclusions about the relative performance of different protocols is difficult, however, since few comparative studies have been conducted. Perhaps the best indicators of how a protocol compares are the design choices that have been made by its developers. For example, a protocol with receiver-initiated error recovery will scale better than one which uses a sender-initiated approach.

Performance should not be the sole discriminator between protocols. Applications may require varying degrees of reliability or resiliency from a protocol. Security considerations may impose other constraints. In all cases, the design choices which have been made will determine how suitable a protocol is for a particular application.

2. Implementation Status

The elements of a protocol's implementation status are taken from (MIST website) and include:

- participants - the individuals and organizations involved in defining and implementing a protocol.
- specification - the status of the specification, if any, written for a protocol.
- availability - the current implementation release information, and who can access it.

- supported platform - platforms which have hosted an implementation of a protocol

Although a number of protocols have been developed to research specific aspects of reliable multicasting, the military should be primarily interested in those which have achieved some degree of commercial acceptance. Protocols implemented more widely than others are likely to spawn additional refinements, and be better tested.

V. RELIABLE MULTICAST PROTOCOLS

A.   INTRODUCTION

Seven different transport layer reliable multicast protocols which run on top of Multicast IP will be described in this chapter. Each will also be classified according to the taxonomy developed in Chapter IV.

The first two protocols, the Tree-based Multicast Transport Protocol (TMTP) and Scalable Reliable Multicast (SRM), were developed for research purposes. Both have been included in this study because their methods of error recovery deserve further examination. The remaining five protocols are "real-world" protocols. The Reliable Multicast Protocol (RMP) is a token-based protocol designed primarily for distributed processing; the Reliable Multicast Transport Protocol (RMTP) provides sequenced, lossless delivery of bulk data; the Reliable Adaptive Multicast Protocol (RAMP) was developed for interactive collaborative applications; the second version of the Multicast Transport Protocol (MTP-2) was built to support teleconferencing applications; the Xpress Transport Protocol (XTP) is a general purpose protocol.

Each of the five "real-world" protocols is relatively mature. Four of the five have been implemented to a certain extent on various platforms. Most are described by a specification; some are formally specified.

Implementation status was not the only criterion for selecting these five protocols, however. Which protocols were studied was also influenced by the type of application each was designed to support. Those chosen were developed to support the types of applications which will be hosted on the tactical internet.

Each protocol will be described before being categorized according to the taxonomy in the previous chapter. If experiments were conducted for a protocol, a summary of the results will be included.

B.   TREE-BASED MULTICAST TRANSPORT PROTOCOL (TMTP)

Collaborative multimedia applications "…involve a large number of participants and are interactive in nature with participants dynamically joining and leaving the applications." (Yavatkar, 1995) Protocols, such as TMTP, which are designed for these

types of applications must incorporate "...carefully designed flow and error control algorithms..." to reduce protocol overhead and processing loads. (Yavatkar, 1995) Those features of TMTP which allow it to conserve network resources merit closer examination, since bandwidth is a scarce commodity in the tactical internet.

1. Protocol Overview

a. *Introduction*

Like the hierarchical error recovery approach of (Lucas, 1995), for TMTP, a specially designated receiver, or "domain manager", is associated with a set of receivers. Unlike the (Lucas, 1995) protocol, however, receivers in these sets are not passive; instead, they are actively engaged in error recovery. Domain managers are responsible for recovering from errors and handling retransmissions to their receivers. They also form a hierarchical control tree rooted at the source by helping a limited number of other domain managers recover from errors. The control tree is built without the direction of a central coordinator; new domain managers join the tree using an "...expanding ring search to locate potential connection points into the control tree." (Yavatkar, 1995)

b. *The Transmission Protocol*

Traffic is sent to all members of a multicast group, including domain managers, using Multicast IP. For parent domain managers (and the source) to reclaim buffer space and implement flow control, child domain managers periodically send acknowledgments to their parent. Limiting the number of children assigned to each domain manager, and only requiring that ACK's be sent periodically reduces the processing burden at parent domain managers and the source. Acknowledgments can be sent immediately after a domain manager receives a packet. This policy allows buffer space to be reclaimed more quickly than a policy which forces a domain manager to delay its acknowledgment until all of its children have received the packet. (Yavatkar, 1995)

c. *Error Control*

"For each domain, its manager maintains a *multicast radius* for the domain, which is the TTL distance to the farthest child within the domain." (Yavatkar, 1995) When a receiver detects an error, it waits a random period of time before

multicasting a NACK to its domain manager and the other receivers in the domain. The scope of the multicast is limited to the domain's multicast radius. A receiver will cancel its NACK if it receives a NACK which corresponds to a packet it has missed. Repairs, which are multicast by domain managers, are sent only to receivers in a domain by restricting their TTL to the multicast radius. Receivers also send periodic ACK's to their domain manager so that it can reclaim buffer space and implement flow control. (Yavatkar, 1995)

d. *Flow Control*

"TMTP achieves flow control by using a combination of rate-based and window-based techniques." (Yavatkar, 1995) The maximum rate at which the sender can transmit, and domain managers can retransmit, is established when the group is created and never changes. "A fixed rate helps avoid congestion arising from bursty traffic and packet loss at rate-dependent receivers...." (Yavatkar, 1995)

"TMTP's window-based flow control differs slightly from conventional point-to-point window-based flow control." (Yavatkar, 1995) Normally, an acknowledgment will advance the lower edge of the sliding window; if the timer set to guard against lost acknowledgments expires, the packet corresponding to the missing acknowledgment is retransmitted. TMTP delays retransmission by making the window a multiple of the acknowledgment timer. Receivers are still required to send acknowledgments during the acknowledgment interval; however, no retransmissions are triggered until all of a window's acknowledgment timers have expired. "This increases the chance of a positive acknowledgment being received and it also allows domain managers to rectify transient behavior..." in their part of the network before it begins to affect other parts of the control tree. (Yavatkar, 1995)

2. Experimental Results

Tests were conducted over a wide area network formed by seven sites in the United States and Europe, with each site acting as a separate domain. All of the experiments "...were conducted using standard IP multicast across the Internet Mbone and thus experienced real Internet delays, congestion, and packet loss." (Yavatkar, 1995)

TMTP was compared to two sender-oriented protocols. The processing load at the sender and domain managers, and the end-to-end delay were compared when the size of the file being transferred was increased from 3 to 30 Kbytes while the number of receivers was held constant at 30. Similar comparisons were made when the file size was held constant and the group size was varied from 5 to 30 members.

Unfortunately, the amount of bandwidth consumed by TMTP was too difficult to quantify. However, the results indicated "...that TMTP generated far fewer retransmissions..." than the sender-oriented protocols, "...and most TMTP retransmissions..." were "...local to a particular domain." (Yavatkar, 1995)

The amount of processing at the sender and domain managers increased only slightly when the file size was increased because TMTP distributes the work of processing acknowledgments and retransmission requests among the nodes of the control tree. Almost no increase in processing load was observed as the group grew because it is more dependent upon the number of domain managers than it is on the number of receivers. Since, for TMTP, error recovery "...proceeds concurrently in different parts of the control tree rather than sequentially...", end-to-end delay rose at a lower rate than the other protocols when either file size or group size was varied. (Yavatkar, 1995)

3.   Evaluation

a.   *Strengths*

"TMTP employs error control techniques from both sender and receiver initiated approaches." (Yavatkar, 1995) The receiver-initiated error recovery exercised by receivers within a domain reduces the processing load on domain managers; hierarchical error recovery distributes the processing burden among the members of the control tree and confines retransmissions to those parts of the network where the loss was experienced; acknowledgments sent by receivers and child domain managers (sender-initiated) allows the buffer space of domain managers and the sender to be reclaimed. By combining both types of approaches, TMTP is able to provide scalable complete reliability.

b.   *Weaknesses*

The hierarchical structure can be a liability, however. If a domain manager leaves the control tree after its last local member has left the group, child domain managers of the departing manager will have to be grafted back into the tree. With the current implementation of TMTP, "...errors that arise during the brief reintegration time might not be correctable." (Yavatkar, 1995)

"The effectiveness of distributed retransmission depends on developing a mechanism to group together multicast receivers with similar error characteristics." (Lucas, 1995) During its expanding ring search a new domain manager chooses a parent in the control tree based on TTL. This simple join mechanism may not guarantee that domain managers become responsible for receivers with similar reception characteristics.

Since "...the control tree is built solely at the transport layer..." some additional end-to-end delay may be introduced by suboptimal routing. (Yavatkar, 1995) However, the control tree is built to facilitate error recovery and flow control, and is not necessarily designed to be optimal. If optimal routing were desired, the mechanism for building and maintaining the tree would become much more complex than the expanding ring search used by the current implementation. (Sudan, 5/21/96 e-mail)

c.   *Summary*

TMTP's error recovery and flow control algorithms seem to be well suited for the tactical internet. The design features of the protocol are summarized in Table 5.1. A blank entry in the "Design Feature" column indicates that either the corresponding category was not addressed by the protocol, or no mention of the feature could be found in the references consulted for this study.

Category	Design Feature
Error Recovery • Responsibility for detecting errors	• domain managers detect packets missed by their child domain managers through acknowledgment timeouts • receivers in a domain detect missing packets using the sequence numbers assigned to the packets
• How errors are signaled	• receivers multicast NACK's (with NACK suppression) to their domain managers and other receivers within their domains • receivers unicast ACK's to their domain managers • child domain managers unicast ACK's to their parent domain manager
• How errors are retransmitted	• domain managers multicast to the local domain (within the multicast radius)
Heterogeneous Receivers	fixed maximum transmission rate for the sender (and retransmission rate for domain managers)
Scalability	• experiment performed with 30 receivers • control tree distributes processing load among the domain managers • receiver-initiated error recovery within domains reduces the processing load at domain managers • hierarchical error recovery confines retransmissions to those areas where the loss occurred
Flow Control	combination of rate-based and window-based techniques • fixed maximum transmission rate for the sender (and retransmission rate for domain managers) • window-based flow control - window partitioned, retransmissions delayed as long as possible
Late-join/Leave	no effort made to coordinate senders and receivers - have to rely on external synchronization method
Fragmentation/Reassembly	
Ordering	single source ordering
Delivery Semantics	
Recovery from Failure	
Prioritization of Traffic	
Group Structure	
Concast or Unicast Capability	can specify either multicast or concast when group established
Other	

Table 5.1. Summary of the Design Features of TMTP.

C. SCALABLE RELIABLE MULTICAST (SRM)

SRM is a reliable multicast framework which has been prototyped in the distributed white board application *wb*. "The algorithms of this framework are efficient, robust, and scale well to both very large networks and very large sessions." (Floyd, 1995)

The focus of the overview of *wb* which begins this section will be how the protocol incorporates application level framing (ALF) principles and how it guarantees

reliability. The section will close with an explanation of the extension to the SRM algorithm which allows error recovery to adapt to changing network conditions.

1.   Protocol Overview

a.   *The wb Whiteboard*

The whiteboard separates the drawing into pages, where a new page can correspond to a new viewgraph in a talk or a clearing of the screen by a member of the meeting. Any member can create a page and any member can draw on any page. Each member is identified by a globally unique-identifier, the Source-ID, and each page is identified by the Source-ID of the initiator of the page and a page number locally unique to that initiator. Each member drawing on the whiteboard produces a stream of drawing operations, or "drawops", that are timestamped and assigned sequence numbers, relative to the sender. (Floyd, 1995)

b.   *Application Level Framing (ALF)*

According to ALF principles, "…as much functionality and flexibility as possible…" should be left to the application. (Floyd, 1995) To comply with these principles, the underlying delivery system of *wb* was designed to provide a relaxed form of reliable multicast. All group members can expect eventual delivery of all data; however, no particular type of ordering is guaranteed. (Floyd, 1995)

Certain behaviors of a protocol are dictated by the application in the ALF model. The application decides how units of data are described (application data units (ADU's)). How bandwidth is allocated, and the priority of different types of data may also be determined by the application. In other words, for reliable multicasting which follows the ALF model, the protocol acts as a framework to which application specific details are added. *Wb* was designed according to the ALF model; the underlying reliable multicast framework is provided by SRM. (Floyd, 1995)

The *wb* ADU is the drawop. Drawops are ordered, not by the multicast protocol, but by the application. Receivers use the timestamps assigned to each drawing operation to determine the order in which the operations appear on the whiteboard. "This course synchronization mechanism captures the temporal causality of the drawing operations at a level appropriate for the application, without the added complexity and delay of protocols that provide guaranteed causal ordering." (Floyd, 1995)

A maximum amount of bandwidth is allocated for each *wb* session; the application prioritizes data within this fixed bandwidth. Packets associated with repairs for the current page are given the highest priority, new data is of a lower priority, and repairs for previous pages are of the lowest priority. (Floyd, 1995)

c.   *The Reliable Multicast Framework*

The receiver-initiated error recovery of SRM is flat receiver-oriented. Repairs and requests for retransmissions are multicast to the entire group. A receiver suppresses its repair request if it hears a request for the same data. A receiver with the requested data can issue a repair if no other receivers are overheard making the same repair.

Each member of a *wb* session sends "…low-rate, periodic, session messages that announce the highest sequence number received from every member that has written on the page currently being displayed." (Floyd, 1995) Besides helping to identify lost data, these session messages are used to determine the current members of a session, and to estimate the distance between participants. This distance is important since it is an element of the algorithms which compute the length of a receiver's request and repair timers. (Floyd, 1995)

"The whiteboard does not require special mechanisms for the detection or recovery from network partitioning." (Floyd, 1995 During a failure, members in the functional part of the partition can continue adding to the whiteboard. After the network has been restored, the normal repair mechanism will recover data lost during the partition. (Floyd, 1995)

2.   Adaptive Loss Algorithm

Simulations reported in (Floyd, 1995) were conducted to analyze the performance of SRM error recovery. Performance was measured by the average number of duplicate repair requests and subsequent repairs for each loss, and the average delay from loss detection to repair. Network topology was found to affect these metrics; adjusting the request and repair timers also influenced performance. To reduce the number of duplicates, and shorten delay, an adaptive loss algorithm was developed which adjusts the request and repair timers based on past error recovery performance. Simulations of the

algorithm showed that it is "…effective in controlling the number of duplicates over a range of scenarios." (Floyd, 1995)

3. Evaluation

SRM is different from other reliable multicast protocols that attempt to satisfy a stricter definition of reliability. Instead, SRM meets a minimum level of reliability while "…leaving more advanced functionalities, whenever needed, to be handled by individual applications." (Floyd, 1995)

The adaptive loss algorithm appears able to improve the scalability of receiver-initiated, flat receiver-oriented error recovery algorithms by reducing the number of duplicate requests and repairs multicast across a network. The algorithm's ability to adjust to changes in network topology, session membership, and congestion may make it more suitable for the dynamic tactical internet than non-adaptive approaches.

The design features of *wb* are summarized in Table 5.2. The "Error Recovery" category describes the characteristics of SRM without the adaptive loss algorithm.

Category	Design Feature
Error Recovery • Responsibility for detecting errors	• receivers detect gaps in sequence numbers • session messages also help identify lost data
• How errors are signaled	receivers multicast repair requests
• How errors are retransmitted	receivers multicast repairs
Heterogeneous Receivers	
Scalability	*wb* tested on a global scale with more than 1000 participants
Flow Control	fixed maximum bandwidth for each session - token bucket rate limiter enforces peak rate
Late-join/Leave	late-join - receiver can issue repair requests to learn of past pages
Fragmentation/Reassembly	
Ordering	application orders based on drawop timestamps
Delivery Semantics	
Recovery from Failure	no special mechanism for detecting or recovering from network partitions
Prioritization of Traffic	application prioritizes traffic
Group Structure	
Concast or Unicast Capability	
Other	

Table 5.2. Summary of the Design Features of *wb*.

D. RELIABLE MULTICAST PROTOCOL (RMP)

"RMP provides a robust foundation for many data replication and groupware applications." (Callahan, 1995) Whereas SRM provides a less stringent form of

reliability to the *wb* application, RMP takes the opposite approach. According to RMP's designer, a distributed application "…should have a robust transport primitive that provides reliable delivery, ordering of messages, and fault-tolerance, thus unburdening the application developer and allowing concentration on the application development itself." (Montgomery, 1994)

1. Protocol Overview

a. *Post-Ordering Rotating Token Model*

RMP is based on the "Post-Ordering Rotating Token" model in which a token is rotated among group members and messages are ordered by the token site after they have been sent. (Montgomery, 1994)

The token list, which contains current members of the token ring, is maintained by all group members, and updated when group members leave the ring, or new members join. Changes to the token list are requested by multicasting a List Change Request (LCR). The token site "…serializes all requests and generates a new token list, referred to as a New List, timestamps the new list, and sends the new list as a token transfer." (Montgomery, 1994) Only one membership change is made for each New List that is created.

How RMP functions can best be explained by an example. Figures 5.1 and 5.2 are snapshots of the operation of the modified rotating token used by RMP. For both Figures, the order of events as they happen on the network is shown under the heading "Event Order" while the global timestamp assigned to these events by the token site can be found under the heading "Imposed Order."
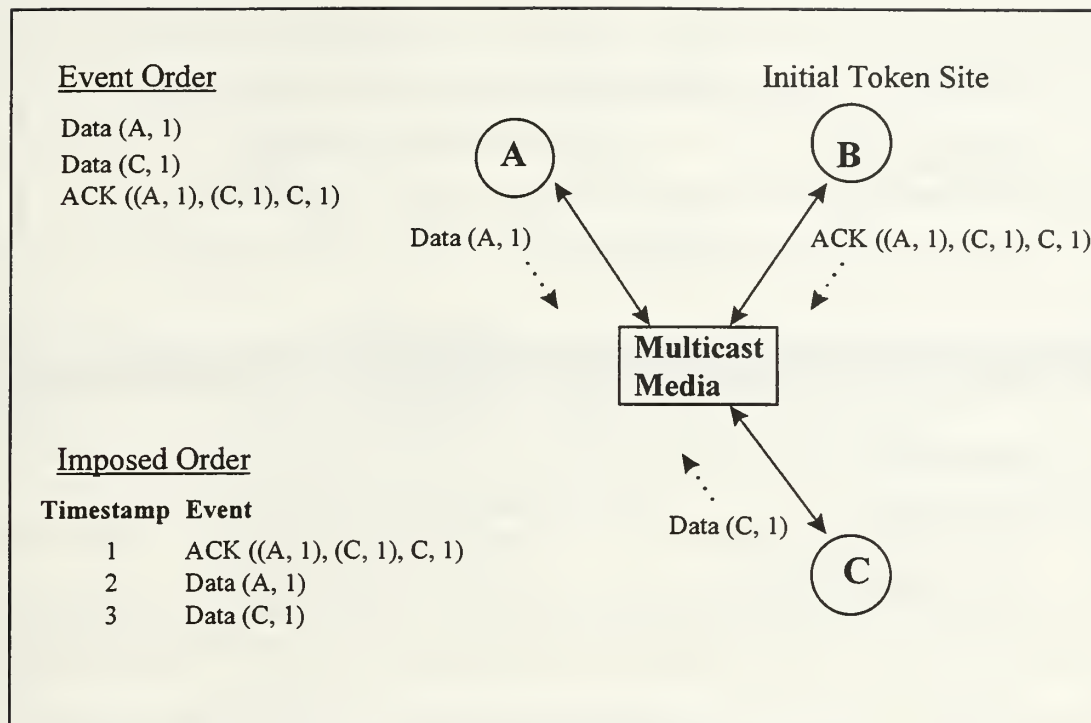
Figure 5.1. First example snapshot of RMP's operation. After (Montgomery, 1994).

At the time of the first snapshot, the token list consists of members A, B, and C; the token site is B. The sequence of events captured by Figure 5.1 is:

- Site A multicasts a message with sequence number 1, represented by Data (A, 1) in Figure 5.1, at roughly the same time that site C multicasts a message with the same sequence number (Data (C, 1)). (Montgomery, 1994)

- The token site (B), receives the message from A before the message from C. Other sites in the group may have received the two messages in the same order as site B, or in the reverse order. To signal to the other group members that the message from A should be ordered before the message from C, the token site multicasts an acknowledgment packet which specifies both the correct ordering of the messages, and the next token site. This acknowledgment packet contains the originating sites and sequence numbers of the messages it is acknowledging ((A, 1), (C, 1)), the next token site (C), and a timestamp (1) to globally order the messages. (Montgomery, 1994)

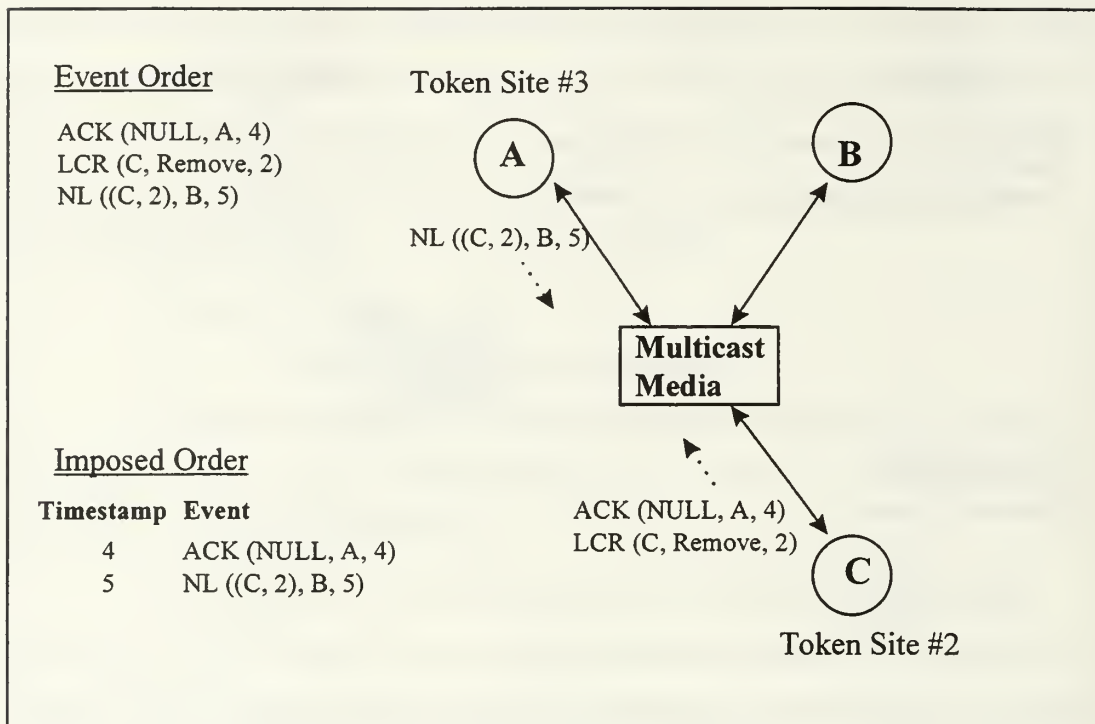Figure 5.2 is the next snapshot of the protocol's functioning.

Figure 5.2. Second example snapshot of RMP's operation. After (Montgomery, 1994).

The token site is now C; the token list has not changed. For site C to accept the token, it must have received all timestamped messages. The sequence of events captured by Figure 5.2 is:

- Because no message is sent within a certain period of time, site C must pass the token. Site C generates a NULL acknowledgment to pass control to the next token site since there are no messages for it to acknowledge. The acknowledgment packet contains the word NULL, the next token site (A), and the global timestamp assigned to the packet (4). Before site A can become the new token site, it must have received all timestamped messages. (Montgomery, 1994)

- Site C asks to leave the ring by multicasting a List Change Request. The sequence number of 2 orders the LCR after the data message sent earlier by site C. (Montgomery, 1994)

- The token site (A) creates a new list, NL ((C, 2), B, 5), that does not contain site C, and multicasts it to the ring. As indicated by (C, 2), the new list was generated by a request from site C. The new list also transfers the token to the

80

next site (B) at global time 5. As before, site B can only accept the token after it has received all timestamped messages. (Montgomery, 1994)

b. Quality of Service (QoS)

Each message is delivered to the application based upon its desired Quality of Service. The different levels of QoS, and how RMP achieves them, are shown in Table 5.3.

QoS Level	Description
Unreliable	Delivery is immediate upon reception. Lost messages are not requested. These messages are not assigned sequence numbers by the sending site.
Reliable	Delivery is immediate upon reception. Lost messages are requested. These messages do receive a sequence number.
Source Ordered	Delivery is after all messages from the same source and with lower sequence numbers have been delivered.
Totally Ordered	Delivery is after all messages with lower timestamps have been delivered.
K Resilient	The same as Totally Ordered but with K-1 passes of the token required as well.
Majority Resilient	The same as K Resilient but with K being equal to $(N + 1)/2$, where N is the size of the ring.
Totally Resilient	The same as K Resilient but with K being equal to N, where N is the size of the ring.

Table 5.3. QoS Levels. From (Montgomery, 1994).

c. Error Recovery

For all but the unreliable QoS, receivers detect losses by noting gaps in the message sequence numbers. "RMP uses a modified SRM Request/Repair policy (as is used in the wb tool)." (Montgomery, 1995) When a receiver detects a missing message, it sets a request timer which is canceled only if the receiver hears a NACK for the same missing message, or if it receives a repair; otherwise, the receiver multicasts a NACK. A receiver with the data requested by a NACK sets a repair timer upon receiving the NACK. If the receiver hears no other repairs before its timer expires, it multicasts the repair to the group.

d. Recovery from Failures

RMP recovers from failures through a fault recovery protocol. The two phases of the protocol are creating and synchronizing the new token list, and committing the list. During the first phase, the site that has detected the failure (Reform Site) polls the other members to determine if they are still active. Those that are active inform the Reform Site of their highest consecutive timestamp and the highest consecutive sequence number they have received from each site in the old token list. Those sites which are

81

missing packets recover them through the normal error recovery procedure and update the Reform Site after receiving repairs. An old token list specifies a minimum number of sites that must remain in a partition after a failure. The new token list must at least meet this minimum before the Reform Site can multicast it to the group. After each member has acknowledged receiving the new list, the Reform Site commits the list and makes itself the token site. Normal operation can then resume. (Whetten, 1995)

e. *Multi-RPC Delivery*

The "Multi-RPC" model of delivery allows processes that are not members of a multicast group to exchange data with group members. This client-server arrangement allows processes to communicate with the ring without incurring the overhead of joining it. RMP achieves further flexibility by accommodating group members that are not multicast capable. (Montgomery, 1994)

f. *Flow and Congestion Control*

"The flow and congestion control policies used by RMP are designed to be orthogonal to the rest of the protocol." (Callahan, 1995) The default policy is a modified sliding window protocol based on the Van Jacobson algorithms used in TCP. (Callahan, 1995)

2. Experimental Results

Experimental tests of RMP have been limited to the LAN environment. The tests reported in (Montgomery, 1994) were run on several SPARCstation 5's (SunOS 5.3) connected to a standard Ethernet LAN. Single sender throughput, aggregate throughput, and packet latency for different levels of QoS were measured. While acknowledging that performance comparisons with other protocols may not be fair since other tests have used different platforms and operating systems, Montgomery (1994) concludes that "...no other reliable multicast protocol has been able to show as high throughput performance as RMP." Single sender throughput approached the theoretical maximum predicted for RMP (9.07 Mbps) for several receivers; throughput decreased slightly as the number of receivers grew to eight. The aggregate throughput is calculated by multiplying the number of destinations by the amount of data sent by all sources. For these tests, senders were also destinations. In all of the test scenarios, the aggregate throughput exceeded the

Ethernet maximum bandwidth of 10 Mbps. Higher quality of service resulted in higher packet latencies. However, latency remained relatively constant as the number of receivers was increased, regardless of the guaranteed QoS. (Montgomery, 1994)

3. Evaluation

a. Strengths

RMP is very flexible. Different qualities of service, from unreliable to totally resilient, can be delivered on a per packet basis. Non-members outside the ring can communicate with group members using Multi-RPC delivery. Also, hosts that are only unicast capable may participate as members of the ring.

Other strengths of RMP are its rotating token model and fault recovery protocol. Knowing when all of the group members have received a packet (message stability) is a by-product of the rotating token model for the higher levels of QoS (Montgomery, 8/21/96 e-mail). The quicker a message stabilizes, the sooner buffer space can be reclaimed. Rotating the token also distributes the processing load among group members. RMP's ability to reconstruct the token ring after a failure would seem to make the protocol well suited for those environments where network partitions are likely.

b. Weaknesses

Although RMP reduces the number of acknowledgments by allowing multiple messages to be acknowledged at once, the protocol overhead still seems to be high. Both messages and acknowledgments are multicast, as are NACK's and repairs. The packets associated with some other less frequent protocol actions, such as joining and leaving a group, are also multicast. In the bandwidth-limited tactical internet, the high protocol overhead generated by RMP may limit its scalability.

Scalability may be limited by the protocol design itself. The total ordering guaranteed by RMP makes it connection-oriented (Montgomery, 8/21/96 e-mail). As long as the protocol must track members, the size of the membership list will bound the receiver set. The eight bit member field currently restricts the number of receivers to 256. (Montgomery, 3/8/96 e-mail).

"RMP is mainly meant for high-availability systems under a somewhat controlled internetwork" such as metropolitan area networks (MAN's) and corporate

intranets. (Montgomery, 8/21/96 e-mail) RMP's flow control mechanism, which degrades performance severely under conditions of moderate packet loss, is the primary reason why the protocol is restricted to low loss environments. Not surprisingly, some experience with RMP has shown that its performance suffers over a WAN. Webcast is a program which shares WWW documents between a group of users; the multicast protocol it uses is RMP. The designers of Webcast noted, in 1995, that "WAN performance is less than perfect." (Webcast website)

Finally, if group membership is expected to change frequently in the tactical internet, RMP's membership policy may slow departures from a group. "When a site removes itself from a ring, it must remain a member of the group until after it has seen and committed the new membership view removing it from the ring." (Whetten, 1995) Furthermore, the site must continue processing repair requests until all the packets it has received from members of the old list are stable. (Whetten, 1995) How long this policy will keep a member involved with a group it has just left is not apparent, but extended involvement may tie up processing resources, and prevent rapid departure from a group.

4. Future Directions

The recently incorporated out-of-sequence (OOS) NACK's, which act much like TCP fast retransmits, may improve RMP's performance in moderate loss environments. According to (Montgomery, 8/21/96 e-mail) the other improvements planned for the protocol are:

- rate based flow control with rate allocation - the rate allocation mechanism will be implemented by RMP, but rate negotiation will be controlled by the application.
- no wait leaves - sites would not have to remain involved with a group after leaving the ring.

Additionally, Montgomery is building a testbed to analyze the performance of RMP in a simulated WAN environment.

5.   Summary

RMP may work best in low loss environments such as LAN's and controlled internetworks, and applications which require high qualities of service, such as those that replicate data and files, may be well served by RMP.

a.   Design Features

The design features of RMP are summarized in Table 5.4.

Category	Design Feature
Error Recovery • Responsibility for detecting errors	• modified SRM Request/Repair policy • receivers detect gaps in sequence numbers
• How errors are signaled	receivers multicast NACK's
• How errors are retransmitted	receivers multicast repairs
Heterogeneous Receivers	different levels of QoS guaranteed on a per packet basis
Scalability	• experiment performed with 8 receivers • protocol design may limit scalability to 256 members • high protocol overhead may limit scalability in a tactical internet
Flow Control	• flow and congestion control orthogonal to the protocol • default mechanism is a modified sliding window protocol
Late-join/Leave	allowed
Fragmentation/Reassembly	performed by other levels of the protocol stack
Ordering	• supports source and total ordering • ordering selectable on a per packet basis
Delivery Semantics	• supports K-Resilient, Majority Resilient, Totally Resilient delivery • delivery semantics selectable on a per packet basis
Recovery from Failure	fault recovery protocol reforms the group
Prioritization of Traffic	
Group Structure	open - processes that are not part of the ring can communicate with the group
Concast or Unicast Capability	
Other	non-multicast capable hosts can be members of the ring

Table 5.4. Summary of the Design Features of RMP.

b.   Implementation Status

From (MIST webpage), the status of RMP's implementation is:

- Participants: Simon Kaplan, University of Illinois, Champagn-Urbana; Todd Montgomery, West Virginia University; Brian Whetten, University of California, Berkeley.

- Specification: "Descriptions of the protocol can be found in RMP Packet Formats, RMP State Specifications, and RMP Flow Control and NACK Policy Specifications." (MIST webpage)

- Availability: The latest C++ version, RMP 1.2+, is available by ftp.

- Supported Platforms: SunOS, IRIX (5.2, 5.3), Ultrix, OSF, Linux, Win32.

E. RELIABLE MULTICAST TRANSPORT PROTOCOL (RMTP)

"RMTP provides sequenced, lossless delivery of bulk data from one sender to a group of receivers." (Lin, 1996) The objective of RMTP is to "…guarantee *complete reliability* at the expense of delay" for applications hosted on a wide area network. (Lin, 1996)

1. Protocol Overview

The multicast delivery tree of RMTP is rooted at a sender; subtrees begin at special receivers called Designated Receivers (DR's) or Acknowledgment Processors (AP's). These DR's are responsible for reliably delivering data to their subtrees.

a. *Session Manager*

"Within RMTP, there is a heavy reliance on the Session Manager (SM) to perform certain functions." (Dismuke, 1995) When a connection is established between the sender and a multicast group, the Session Manager provides all participants with the connection parameters. They include: send and receive window sizes, the multicast retransmission threshold, data packet size, and other parameters which affect protocol performance. The Session Manager also notifies the application when a receiver voluntarily or involuntarily leaves a multicast group. Finally, the Session Manager plays an important role in congestion avoidance by choosing and setting the maximum transmission rate. (Lin, 1996)

b. *Error Recovery*

The sender divides the data to be transmitted into fixed packets based on the data packet size chosen by the Session Manager. Each packet is assigned a sequence number before being multicast to the group. DR's learn of missing packets from acknowledgments unicast to them by receivers in their region. DR's inform their upstream counterparts of missing packets through the same acknowledgment mechanism. (Lin, 1996)

The acknowledgments sent by receivers consist of a sequence number and

a bitmap. The ACK sequence number corresponds to the lower edge of the receive window and is equal to one more than the highest numbered consecutive packet correctly received. The bitmap shows which messages are in the receive buffer. (Lin, 1996) For example, an ACK with a sequence number of 20 and a bitmap of 01100111 indicates that packets with sequence numbers less than 20 have been correctly received. The zeros in the first, fourth, and fifth bits also indicate that sequence numbers 20, 23, and 24 have been lost.

To avoid inefficiencies, a receiver's ACK interval is dynamically adjusted based on the round trip time between the receiver and its AP. An excessive ACK interval will lengthen the time between retransmissions, thereby delaying the delivery of packets to the application. If ACK's are sent too frequently "...the AP may end up retransmitting the same packet multiple times without knowing if the first retransmitted packet was received correctly by the receivers." (Lin, 1996)

Repairs may either be multicast or unicast by the DR. The number of requests for a particular packet during a retransmission interval is tabulated by the DR. If these requests exceed the multicast transmission threshold, the repair is multicast to the receivers in a region; otherwise, the missing packet is unicast to each receiver that has requested a retransmission. To eliminate the delay introduced by this feature, the retransmission interval can be reduced to zero so that repairs are sent immediately. (Lin, 1996)

c. Two-level Caching

RMTP guarantees that receivers, including those that join late, will correctly receive all the data transmitted during a session. To meet this requirement, the sender and DR's must buffer all of a session's data. The amount of data equal to the cache size specified by the Session Manager is stored in memory; the rest is stored on disk. "Although keeping all the transmitted data increases the size of state information maintained at each AP, the state size is not a function of the number of multicast participants..."; instead, it is a function of the size of the file transmitted during a session. (Lin, 1996)

d. *Immediate Transmission*

Late joins are able to catch up to their peers using the immediate transmission feature. This feature uses a special ACK packet to notify the DR that a late join is missing data. The DR responds by immediately unicasting the requested data to the receiver. This feature is also used by receivers that have fallen behind because of congestion or network partitions. (Lin, 1996)

e. *AP Selection*

"Although the DRs are chosen statically for a multicast group, a receiver uses a mechanism to dynamically choose a DR as its AP." (Lin, 1996) The DR chosen to be a receiver's AP is the one closest to the receiver. An AP is selected with this mechanism at connection establishment, and when an established AP fails. (Lin, 1996)

f. *Flow Control and Congestion Avoidance*

RMTP "...uses windowed flow control with congestion avoidance to avoid overloading slow receivers and links with low bandwidth." (Lin, 1996) The progress of the slowest receiver determines the send window size. The slowest receiver is the one which has sent the smallest sequence number in its acknowledgments during a fixed interval. This sequence number becomes the lower edge of the send window. RMTP attempts to minimize congestion based losses by reducing the transmission rate when congestion is detected. Possible network congestion is signaled by the number of retransmission requests during a fixed interval. Additional losses are avoided by allowing receivers to set an upper bound on the transmission rate (through the Session Manager). (Lin, 1996)

2. Experimental Results

The network configuration used to test RMTP's performance was an Internet WAN consisting of 18 receivers located in five geographic areas. One of the sites was located in Taiwan. Each geographic area was served by one DR. The three test scenarios were: multicasting to a LAN, multicasting to the domestic receivers, and multicasting to all receivers. For the LAN test, throughput, number of retransmissions, and the number of slow starts were measured. For the other tests, the number of duplicates were also counted.

The most significant results were observed in the WAN tests. For both WAN scenarios, the number of retransmissions showed how important DR's were in "…caching received data, processing ACKs, and handling retransmissions." (Lin, 1996) The small variation in throughput demonstrated that RMTP adjusts to receivers in various network environments. Also, the low number of duplicate packets showed how effective the algorithm for calculating the ACK interval was. (Lin, 1996)

These experiments also highlighted the demands placed on protocols by the WAN environment. The number of slow starts increased significantly as the test was shifted from a LAN to the domestic WAN. Further increases were observed when the international link was included. The lower throughput seen during the WAN tests may have been caused both by the lower bandwidth of the wide area links, and the increased congestion as suggested by the rise in the number of slow starts.

3.   **Evaluation**

a.   *Strengths*

RMTP seems to be well suited for delivering bulk data reliably to those applications on the tactical internet which are tolerant of delays. Its flow control and congestion avoidance mechanisms will help RMTP adapt to the heterogeneous receivers and different bandwidths expected on the tactical internet. By localizing ACK's and retransmissions, RMTP conserves bandwidth in those areas of the network which have not experienced loss. Other protocol overhead is expected to be small and is configurable (Paul, 6/24/96 e-mail). Sender throughput will not be slowed by protocol processing since the responsibility for processing ACK's and sending retransmissions is distributed among the sender and DR's.

RMTP's ability to choose between sending repairs unicast or multicast has two positive effects. First, the number of retransmissions sent to receivers that don't need them is reduced. Secondly, the effectiveness of RMTP's hierarchical error recovery scheme may no longer depend on grouping receivers with similar error characteristics as was stated in (Lucas, 1995).

b. *Weaknesses*

In a tactical internet, packets may be dropped because of poor quality links as often as they are dropped because of congestion. Although an increased number of retransmission requests may be an indication of either situation, different responses are required in each case. If more retransmission requests signal network congestion, the sender's transmission rate should be reduced, as it is in RMTP. However, if the number of retransmission requests is an indication of "dirty" links, the transmission rate should be increased to compensate for dropped or corrupted packets. By only associating an increased number of retransmission requests with congestion, RMTP risks responding inappropriately when these numbers indicate poor quality transmission links.

On one hand, the decision to adjust the sender's window based on the progress of the slowest receiver prevents less capable receivers from being overrun. On the other hand, this policy does not fully exercise the processing capacity of faster receivers.

4. Summary

a. *Design Features*

The design features of RMTP are summarized in Table 5.5.

Category	Design Feature
Error Recovery	
• Responsibility for detecting errors	receivers detect gaps in sequence numbers
• How errors are signaled	• ACK's include a sequence number and a bitmap indicating which packets have been correctly received • receivers unicast ACK's to their AP • AP's unicast ACK's to their upstream AP's; AP's immediately downstream from the sender ACK to the sender
• How errors are retransmitted	AP's either unicast or multicast repairs to their receivers depending on how many requests are received for a missing packet
Heterogeneous Receivers	slow receivers determine the upper bound on the transmission rate
Scalability	• experiment performed with 18 receivers located at different sites in the U.S. and Taiwan • state information maintained by each participant is independent of the number of participants • processing load is distributed among the sender and AP's • protocol overhead is expected to be small and is configurable • ACK's and repairs are confined to regions where the loss occurred • in cases where few receivers are affected by a loss, unnecessary repairs are avoided by unicasting the repairs
Flow Control	• windowed flow control • users can set an upper bound on the sender's data transmission rate • slow start congestion window reduces transmission rate when congestion is experienced
Late-join/Leave	• receiver can join a session late and still receive all data reliably because of two level caching scheme • reliable delivery not guaranteed to receivers that leave early
Fragmentation/Reassembly	sender divides data to be transmitted into fixed-size packets
Ordering	single source ordering
Delivery Semantics	
Recovery from Failure	reliability not guaranteed during network failures - application notified by Session Manager
Prioritization of Traffic	
Group Structure	
Concast or Unicast Capability	
Other	

Table 5.5. Summary of the Design Features of RMTP.

b. Implementation Status

From (MIST webpage), the status of RMTP's implementation is:

- Participants: John Lin, Purdue University; Sanjoy Paul, Bell Laboratories; Krishan Sabnani, Bell Laboratories.

- Specification: Formal specification is available from sanjoy@bell-labs.com. (Dismuke, 1995) is another formal specification of the protocol.

- Availability: AT&T proprietary.

- Supported Platforms: SunOS, Solaris 2.x, UNIX SVR3, UNIX SVR4 (RMTP/UDP/IP), NCR UNIX MP-RAS 3.0 (RMTP/IP in kernel).

F.   RELIABLE ADAPTIVE MULTICAST PROTOCOL (RAMP)

RAMP was designed for collaborative-interactive applications hosted on "…all-optical, circuit-switched, gigabit…" networks. (Koifman, 1996) However, "…RAMP's design is also relevant for the next generation of packet switched networks." (Koifman, 1996)

1.   Protocol Overview

a.   Introduction

Collaborative-interactive applications such as shared whiteboards or data conferencing applications require low latencies and high throughput. For these types of applications, the number of participants is also expected to be relatively small. (Koifman, 1996)

Three characteristics of high performance networks will make packet losses more the result of overflows in receiver buffers than of network congestion. The low bit-error rates of these networks will lead to fewer retransmissions, thereby reducing network congestion. Optical crossbar switches, which do not store and forward packets, will further reduce the likelihood of congestion. Finally, whereas most receivers are generally faster than conventional networks, gigabit networks may outstrip the processing capability of some receivers. (Koifman, 1996)

The error recovery approach adopted by RAMP is designed to provide the low latencies demanded by collaborative-interactive applications, and to eliminate unnecessary processing at both senders and receivers. Of the two data delivery modes which can be selected by the sender, the Burst mode scales to the group sizes expected in

collaborative-interactive applications; the Idle mode accommodates larger receiver sets. (Koifman, 1996)

b. *Burst Mode*

For both modes, a burst of data is a series of packets which follow each other within some specified interval. When the time between two consecutive packets exceeds this interval, the first packet marks the end of a burst while the second packet marks the beginning of the next burst. (Koifman, 1996)

Figure 5.3 shows how data flows from the sender in the Burst mode. In this example, a connection is initiated with a Connect request (*active* open). The "A" in the Connect request indicates that a receiver must acknowledge the request with an Accept response if it wishes to join the group.

In Burst mode, the beginning of each burst is marked by a data packet with the ACK flag set. After a burst, a single Idle message is sent to signal that the sender will remain silent until the beginning of the next burst. Receivers that have chosen to guarantee reliability must respond to the sender's ACK flag by returning the sequence number of the data packet which has the ACK flag set. If the sender does not receive an ACK from the receiver, it retransmits the data message. If no response is received after several retransmissions, the sender closes its control channel to the failed receiver. (Koifman, 1996)

The sender is responsible for detecting and acting on failed connections in Burst mode. Although the amount of network traffic is less than the traffic in the Idle mode, the sender must process acknowledgments from all the receivers at the beginning of each burst. (Koifman, 1996)
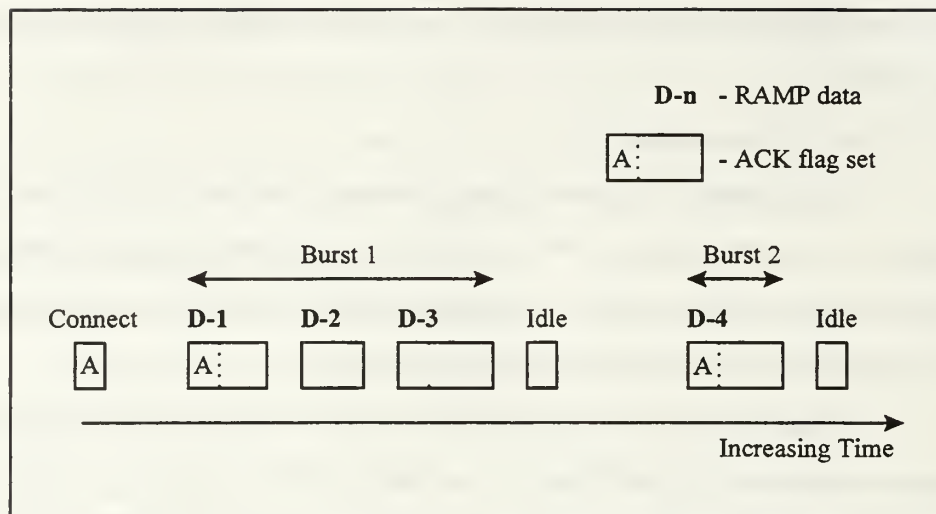
Figure 5.3. Data Flow from the Sender in Burst Mode. After (Koifman, 1996).

c. *Idle Mode*

For the Idle mode (Figure 5.4), the acknowledgment flag is not set in the data packet marking the beginning of a burst; neither is a single Idle message transmitted after the last packet in a burst. Instead, a series of Idle messages is multicast between bursts, each within a fixed time interval. If neither a data packet or an Idle message is received within this fixed interval, a receiver unicasts a Resend message to the sender. If the sender does not respond to a Resend message, the receiver closes its control channel to the sender. Receivers also periodically unicast Idle messages to the sender to verify the health of the control channel. If an Idle message from a receiver is lost, the sender closes the connection to that receiver. (Koifman, 1996)

The receiver is responsible for detecting and acting on failed connections in Idle mode. Although the periodic Idle messages consume more bandwidth than the protocol overhead of the Burst mode, they relieve the sender of the burden of processing acknowledgments. (Koifman, 1996)
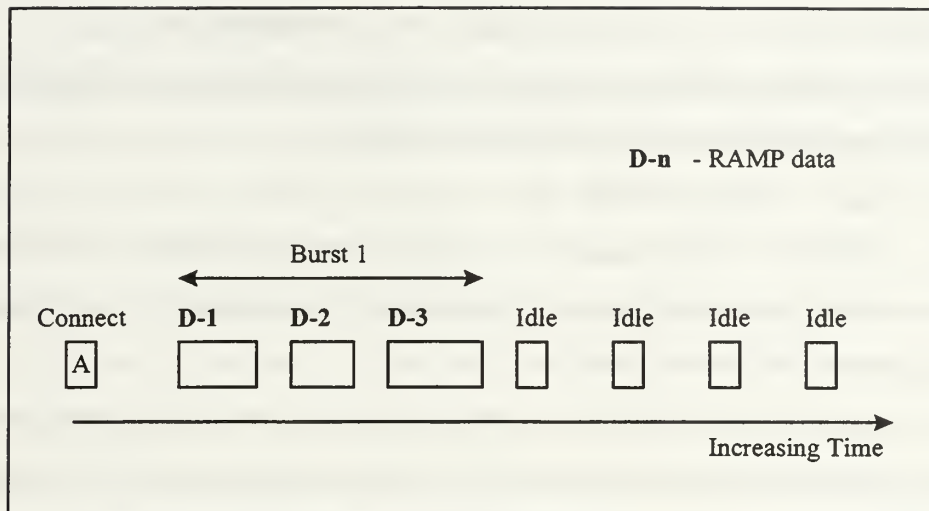
Figure 5.4. Data Flow from the Sender in Idle Mode. After (Koifman, 1996).

d.   *Error Recovery*

Upon detecting a gap in the sequence numbers of arriving data packets, a receiver unicasts a NACK to the sender. The application decides whether missing data is unicast or multicast. (Koifman, 1996)

e.   *Connection Model*

"All data flows from a sender to the receivers over the data channel using a combination of multicast and unicast, and all control traffic flows from the receivers to the sender over a unicast control channel." (Koifman, 1996)

"RAMP can be described as a connection-oriented, reliable stream service." (Koifman, 1996) The sender's explicit knowledge of the receiver set facilitates circuit setup and allows immediate termination of traffic to a group after the last member leaves, but limits scalability. (Koifman, 1996)

f.   *Unreliable Delivery*

RAMP can also provide "...two types of unreliable data delivery to support applications where unreliable delivery is acceptable and appropriate." (Koifman, 1996) When receivers do not request repairs or send acknowledgments, and senders do not process control traffic from the receivers, complete unreliability is provided. (Koifman, 1996)

The second type of unreliable data delivery allows some receivers to guarantee reliability while others guarantee none. For this delivery model, the sender

95

supports reliable delivery by processing acknowledgments and responding to retransmission requests; receivers can operate reliably by requesting retransmissions, or unreliably by ignoring lost messages. (Koifman, 1996)

g. *Flow Control*

RAMP's flow control is based on the number of NACK's which each receiver transmits to the sender during a fixed interval. The sender's rate is adjusted to the slowest receiver. (Koifman, 1996)

A receiver which delivers packets to the application slower than the transmission rate will not generate any NACK's until its buffers overflow. To prevent this, RAMP causes the receiver to intentionally drop packets when its queue becomes greater than 80% full. The NACK's generated by such a policy throttle the transmission rate before a receiver's buffers overflow. (Koifman, 1996)

2. Experimental Results

RAMP's performance as a reliable multicast, unicast, and concast protocol was tested on six UNIX workstations connected to a twisted-pair multi-segmented Ethernet LAN. (Koifman, 1996)

For the reliable multicast scenario, the measured effective throughput for 5 receivers approached the theoretical throughput. When packets were intentionally dropped at all receivers, RAMP's unicast retransmission reduced the effective throughput. (Koifman, 1996)

Although the aggregate throughput dropped only slightly as the number of senders was increased in the reliable concast scenario, this test highlighted the need to develop a "…more scalable flow control approach…for high volume concast communications…." (Koifman, 1996)

3. Evaluation

a. *Strengths*

The clear focus on the requirements of collaborative-interactive applications in the design of RAMP may make it the best protocol for this type of application.

b.   *Weaknesses*

The protocol overhead of both modes of data flow could consume relatively high amounts of bandwidth depending on the size and frequency of data bursts. In Burst mode, protocol overhead consists of a single Idle message at the end of each burst, and an acknowledgment from the receiver at the beginning of each burst. Large, less frequent data bursts will lessen the impact of this overhead. In Idle mode, bandwidth is consumed by the Idle messages sent between bursts by both the sender and the receivers. Frequent bursts will minimize the number of Idle messages. Frequent, long data bursts will further reduce the proportion of bandwidth due to protocol overhead.

The same criticism made of RMTP's flow control mechanism can also be made of RAMP's flow control. By assuming that an increased number of NACK's signals packet loss at the receivers instead of in the network, RAMP risks responding inappropriately in a tactical internet where link losses and congestion may generate the most NACK's.

4.   Summary

a.   *Design Features*

The design features of RAMP are summarized in Table 5.6.

Category	Design Feature
Error Recovery • Responsibility for detecting errors	• Burst mode - sender is responsible for identifying and acting on failed connections • Idle mode - receiver is responsible for identifying and acting on failed connections • in both modes, receivers detect errors
• How errors are signaled	• in Burst mode receivers send periodic ACK's in response to ACK flags set by the sender • in Idle mode, receivers issue Resend message if no data packets or Idle messages arrive within an interval • in both modes, NACK's are sent after gaps in sequence numbers are detected
• How errors are retransmitted	retransmissions are either unicast or multicast (application selectable)
Heterogeneous Receivers	• both senders and receivers can act unreliably • senders can act reliably while receivers act either reliably or unreliably

Scalability	• experiment performed on multi-segmented Ethernet LAN with 6 workstations
	• Burst mode
	  • designed for small receiver groups (< 100)
	  • ACK's periodically sent from receivers in response to ACK flags set by the sender
	• Idle mode
	  • designed for large receiver groups
	  • Idle messages multicast from sender to all receivers
	  • receivers periodically send Idle messages to sender
	• sender maintains explicit knowledge of receiver set

Flow Control	• flow control designed for packet switched networks • transmission delay factor calculated for each receiver based on the number of NACK's
Late-join/Leave	• fast joining or leaving - can leave or join at any point in a session • late receivers cannot request retransmission of messages with earlier sequence numbers
Fragmentation/Reassembly	• data flow is divided logically into bursts • large messages can be fragmented into transfer units of 8000 bytes
Ordering	
Delivery Semantics	
Recovery from Failure	
Prioritization of Traffic	
Group Structure	
Concast or Unicast Capability	both reliable unicast and reliable concast capability provided
Other	

Table 5.6. Summary of the Design Features of RAMP.

b. *Implementation Status*

From (MIST webpage), the status of RAMP's implementation is:

- Participants: Alex Koifman, TASC; Steve Zabele, TASC.

- Specification: Initially described in IETF Network Working Group RFC-1458; enhanced version described in (Koifman, 1996).

- Availability: Contact Steve Zabele (gszabele@tasc.com) to license either the libraries or source code.

- Supported Platforms: IRIX, SunOS, Solaris.

G. MULTICAST TRANSPORT PROTOCOL (MTP - 2)

The Multicast Transport Protocol (MTP) is specified in (Armstrong, 1992). Several of MTP's weaknesses have been addressed by a separate development team; their protocol, named MTP-2, is designed for teleconferencing applications.

1. Protocol Overview

Like its predecessor, MTP-2 is "…based on the notion of a multicast 'master' which controls all aspects of group communications." (Braudes, 1993)

a. *Sending Data*

Figure 5.5 depicts the operation of the MTP protocol for two members of a multicast group (A, B) which act as both a producer and a consumer. Transmissions sent from A and the master are represented by solid lines; transmissions from B are represented as dashed lines.

Producers cannot send data without a token. To obtain a token, producers unicast a token request to the master. The unicast response specifies a message-number. "Producers mark all packets of the message for which the token was granted with the message-number; they return the token implicitly with the final packet of the message…." (Bormann, 1994) In Figure 5.5, producer A is assigned message-number 1 while producer B is assigned message-number 2. Data packets are multicast to all members of the group, including the master. Packet sequence numbers increase monotonically within the same message number, as shown in Figure 5.5.

b.   *Atomicity and Ordering*

The empty packet sent by the master includes the message acceptance record which gives the status of the 12 most recent messages. Although message-number 1, packet 1 was lost before arriving at group member B, it was received by the master. Since all packets of messages 1 and 2 have arrived at the master, they are accepted by the master. Messages accepted by the master can be delivered to the application in the same order as their message-numbers. Group members have the option of disabling atomicity so that messages are delivered without having to wait for them to be accepted by the master. (Bormann, 1994)

c.   *Error Detection and Recovery*

When receivers detect an error, they unicast a NACK to the producer. Repairs are multicast to the group. Complete reliability is not guaranteed, though, since producers are only obligated to retain packets for a certain period of time. (Bormann, 1994)
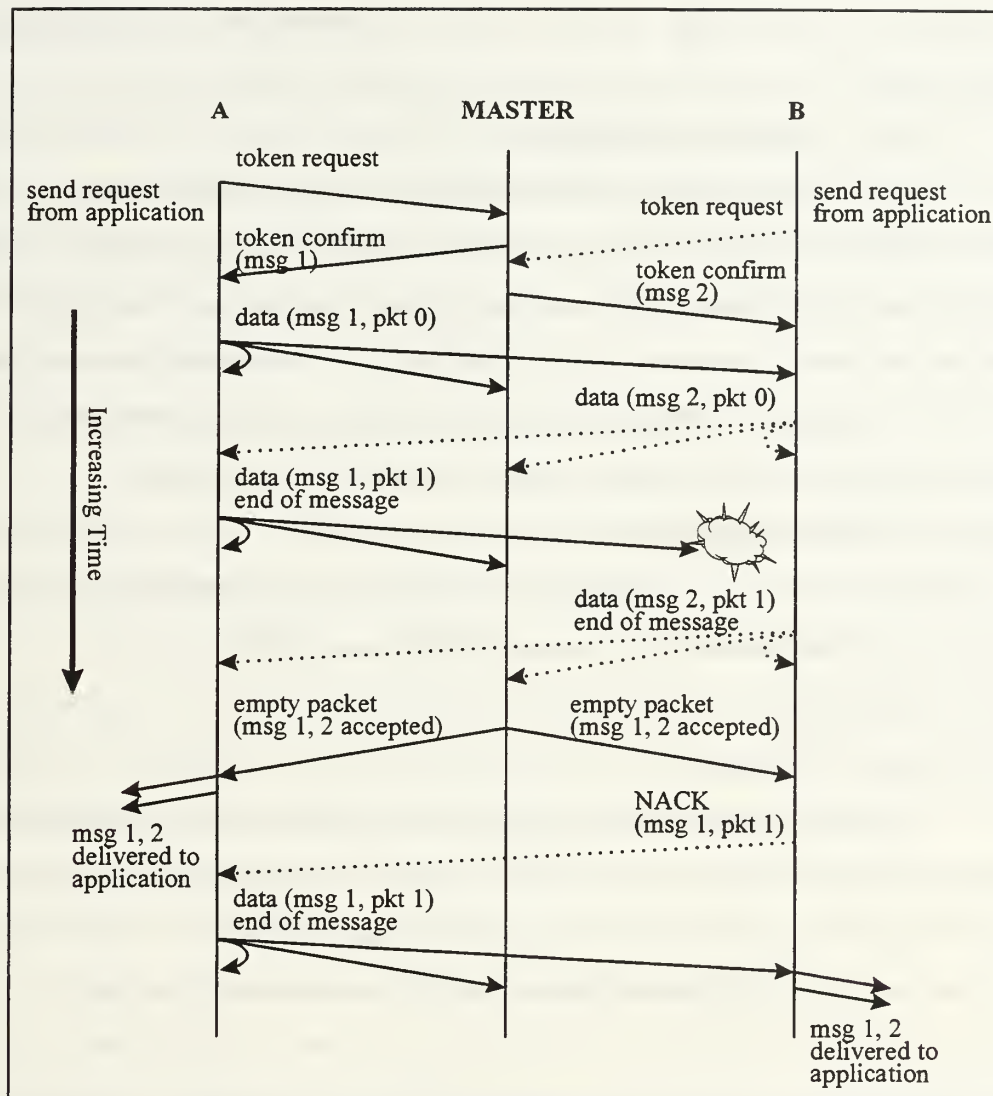
Figure 5.5. Example Operation of the MTP-2 Protocol. After (Bormann, 1994).

d.   *Master Loss and Migration*

The suspected loss of the master is confirmed by the receivers sending it a special packet. If the master does not respond, the members assume that it has failed and elect a new master. Much like failure recovery in RMP, the new master "…accumulates information about the status of all active messages…from all responding…members." (Bormann, 1994)

A procedure similar to loss recovery can be followed to move the master to another host. Master migration may be necessary if the current host is overloaded, or if the connection to the master is congested. To reduce the number of token request and

confirm packets, the master might also move to the sole producer for a multicast group. (Bormann, 1994)

e. *Parameter Adjustment*

Transmission time is divided into equally spaced intervals called heartbeats. When no data is being sent, the master multicasts an empty packet for each heartbeat. To reduce the impact of this overhead, the heartbeat can be slowed when producers are quiescent. The amount of time that producers must retain data may also be adjusted. "If frequent packet loss is encountered..." a higher retention period may "...increase the likelihood of successful (re)transmission." (Bormann, 1994)

2. Experimental Results

MTP-2 was tested on 27 workstations connected to an Ethernet LAN. No significant results from this test were discussed in (Bormann, 1994).

3. Evaluation

a. *Strengths*

For fairly reliable networks, the protocol overhead seems small. For each message sent, the protocol adds a unicast token request and confirm packet. However, for one-to-many multicasting, the master can be migrated to the producer to eliminate this network overhead. The negligible 24 bit message acceptance record is included in all packets sent by the master and in all packets sent by producers. Empty packets multicast for each heartbeat when data is not being transmitted also consume some bandwidth, but the frequency of the heartbeat can be slowed during these periods to reduce overhead.

Being able to recover from a failure of the master is important if MTP-2 is to operate over the tactical internet. Master migration also keeps the protocol viable in the face of changing network or host conditions.

A number of other features make MTP-2 more versatile than other protocols. "MTP-2 defines a separate unicast channel per producer-consumer pair." (Bormann, 1994) Token requests can be prioritized; the master processes requests in order of their priority. Messages can also be prioritized. (Bormann, 1994)

b. *Weaknesses*

The receiver-initiated/sender-oriented error recovery of MTP-2 will probably not scale. Although unicast NACK's are less expensive than multicast NACK's, retransmission requests for a missing packet may be made by more than one receiver, leading to duplicate multicasts of the same packet. (Bormann, 1994) confirms that MTP-2 does not perform well in the type of network conditions expected in the tactical internet: "Small numbers of packet losses can be repaired without too much loss of throughput; extremely lossy multicast distribution trees are not useful...."

Another possible weakness of the protocol is that the size of the message acceptance record limits "...the number of concurrent messages that can be pending (in circulation) to 12." (Bormann, 1994)

Finally, MTP-2 does not guarantee complete reliability. Because producers discard data after a certain retention period, complete reliability cannot be guaranteed. Although this relaxed reliability may not be tolerable to some applications, it may be acceptable for the teleconferencing applications which MTP-2 targets.

4. Summary

a. *Design Features*

The design features of MTP-2 are summarized in Table 5.7.

Category	Design Feature
Error Recovery	
• Responsibility for detecting errors	receivers detect gaps in sequence numbers
• How errors are signaled	receivers unicast NACK's to the producer
• How errors are retransmitted	repairs are multicast to the group
Heterogeneous Receivers	atomicity can be disabled on a per message basis
Scalability	• testing conducted on 27 workstations connected to an Ethernet LAN • receiver-initiated/sender-oriented error recovery
Flow Control	
Late-join/Leave	• immediate late join allowed • late joins do not request packets missed before join
Fragmentation/Reassembly	
Ordering	total ordering
Delivery Semantics	atomicity supported
Recovery from Failure	multicast group can be reformed after the "master" is lost
Prioritization of Traffic	• token requests can be prioritized • messages can be prioritized within streams
Group Structure	
Concast or Unicast Capability	• separate unicast channel defined for each producer-consumer pair • unicast and multicast messages are not synchronized
Other	heartbeat and retention parameters can be dynamically adjusted

Table 5.7. Summary of the Design Features of MTP-2.

b. Implementation Status

From (MIST webpage), the status of MTP-2's implementation is:

- Participants: Carsten Bormann, Universität Bremen; Jörg Ott, Nils Seifert, Technische Universität Berlin.

- Specification: The MTP specification is IETF Network Working Group RFC-1301 (Armstrong, 1992); the MTP-2 specification is only available in German.

- Availability: MTP-2 can be obtained via the FTP server at the Technical University of Berlin.

- Supported Platforms: SunOS 5 / Solaris 2, SunOS 4.1.3, Linux 1.2.8.

H. XPRESS TRANSPORT PROTOCOL (XTP)

"The Xpress Transport Protocol (XTP) is a high-performance transport protocol designed to meet the needs of distributed, real-time, and multimedia systems in both unicast and multicast environments." (Atwood, 1996)

1. Protocol Overview

a. Connection Model

XTP multicast data flow is simplex. Data packets are delivered from a transmitter to a set of receivers; control traffic flows in the opposite direction. Multiple instances, or contexts, of XTP can be active at an endpoint. At receivers, incoming packets must be mapped to a context by consulting a translation map. Senders can take advantage of an optimization which allows packets to be matched to their context without performing a lookup in the translation database.

b. Multicast Group Management

The multicast group is managed by the user based on information which the transmitter maintains about its receivers. The user specifies "…how the initial group of active receivers is compiled, and the criterion for admission…" once the multicast group is established. (XTP Forum, 1995) The user also decides what behavior warrants removal from the group. Finally, information about the receivers helps the application enforce the desired level of group reliability. (XTP Forum, 1995)

The XTP specification only recommends the type of information that the transmitter should maintain about its receivers; it does not "…specify the form and content of this data structure…." (XTP Forum, 1995) The range of group policies that can be enforced depends on what the user requires the transmitter to know about its receivers; the transmitter learns about its receivers by periodically soliciting control packets from them.

To join an in-progress session, a receiver multicasts a JOIN packet to the group. Only the sender is allowed to answer a join request. The receiver begins receiving new data beginning with the sequence number in the JOIN packet returned by the sender. (XTP Forum, 1995)

The three ways that a member can leave a multicast group are: voluntary exit, forced exit, and silent exit. A receiver voluntarily leaving a multicast group must wait in a "zombie state" after asking to leave the group. The receiver is removed from the group after a timer expires. A forced exit, on the other hand, is initiated by the

transmitter. Receivers are also removed from the group by the transmitter if they do not respond to a synchronizing handshake (silent exit). (XTP Forum, 1995)

XTP can support different levels of group reliability. The user may require that all members of the group "...remain alive and current..." for the group to survive. (XTP Forum, 1995) Alternatively, the user may require that "...at least k receivers be operational for the group to continue...." (XTP Forum, 1995) The application may also specify that only certain receivers be fully reliable. In each instance, the transmitter's explicit knowledge of the receiver set permits the application to implement some level of group reliability. (XTP Forum, 1995)

c. *Error Control*

"The same error control procedures defined for unicast associations are available in multicast mode." (XTP Forum, 1995) Normally, receivers wait until the sender asks for updates to inform it of missing packets. However, an option is available which allows receivers to notify a sender immediately after detecting a loss. Control packets indicating lost data contain the highest consecutive sequence number received, and the missing spans of sequence numbers. Repairs are multicast to the group; retransmissions may either be selective or go-back-N. The sender may disable error control, but, even so, it may still ask receivers for control information to regulate data flow. (XTP Forum, 1995)

A synchronizing handshake "...serves to establish a point in the association when each participant has up-to-date status information." (XTP Forum, 1995) Either the transmitter or the receiver can initiate a synchronizing handshake.

d. *Flow and Rate Control*

"The multicast transmitter obeys the same rules for flow control, rate control, and error control as a unicast sender." (XTP Forum, 1995) To regulate these aspects of group communications, the multicast transmitter must collect control information from all receivers. "How these values are aggregated from the receiver group is implementation, and possibly application, specific, and is not defined by XTP." (XTP Forum, 1995)

2. Experimental Results

Performance tests of XTP have been conducted on earlier releases of the protocol. Although the current release of XTP has been implemented by several organizations, no comparative studies of its performance are available in the literature.

3. Evaluation

a. Strengths

XTP is perhaps the most mature of the protocols evaluated. Development of XTP began in 1987; many revisions have been released since then. A number of organizations, including the Navy, have implemented XTP. The XTP Forum has drawn upon the experiences of this user base to make improvements to the protocol; further improvements are expected in the future.

b. Weaknesses

The scalability of XTP will be limited by its method of error recovery. Like other sender-initiated error recovery approaches, the transmitter may be unable to maintain state for large receiver groups and sender throughput may be slowed by control packet processing. Multicasting repairs to the group will also consume unnecessary bandwidth on links leading to those receivers which do not need repairs.

4. Summary

a. Design Features

The design features of XTP are summarized in Table 5.8.

Category	Design Feature

Error Recovery
    • Responsibility for detecting errors
	• receivers detect gaps in sequence numbers
	    • normally, receivers respond with control packets when the sender asks for status
	    • option for receivers to immediately inform sender of an error
	• worst case values taken from set of control packets

• How errors are signaled	receivers unicast ECNTL packets to the sender specifying the highest consecutive sequence number received, and the missing sequence number spans
• How errors are retransmitted	• repairs are multicast to the group • sender can also retransmit to a specific receiver • retransmission can be selective or go-back-N
Heterogeneous Receivers	different levels of reliability can be supported
Scalability	• receiver-initiated/sender-oriented error recovery • senders maintain explicit knowledge of receiver set • sender may have to maintain at least minimal records of all receivers that have ever been part of an association
Flow Control	• control packets sent from receivers which update flow and rate control values • how these values are aggregated is not defined by XTP
Late-join/Leave	• late joins receive traffic beginning with the sequence number in the JOIN packet • receivers are allowed to voluntarily leave the group
Fragmentation/Reassembly	
Ordering	single source ordering
Delivery Semantics	supports atomicity, k-reliable delivery, unreliable delivery
Recovery from Failure	
Prioritization of Traffic	
Group Structure	
Concast or Unicast Capability	
Other	

Table 5.8. Summary of the Design Features of XTP.

b. *Implementation Status*

From (MIST webpage), the status of XTP's implementation is:

- Participants: Members of the XTP Forum.

- Specification: (XTP Forum, 1995) is the XTP 4.0 Specification. (Atwood, 1996), the Addendum to the XTP 4.0 Specification, describes changes made to the 4.0 multicast mechanism.

- Availability: The Sandia implementation of XTP is available from strayer@ca.sandia.gov. Other implementations are available through the XTP Forum.

- Supported Platforms: Contact the XTP Forum to learn of platforms that have hosted XTP implementations.

VI. CONCLUSIONS AND RECOMMENDATIONS

A.   CONCLUSIONS

For applications which require communicating from a sender to many receivers, multicasting is more efficient than other alternatives. The bandwidth saved by multicast protocols make them particularly valuable in the tactical internet.

1.   Multicast Routing Protocols

Which protocols are best suited for the tactical internet is largely determined by how well they adapt to the unique requirements of this environment. Table 6.1 summarizes the behaviors that multicast routing protocols operating in a tactical internet must exhibit.

Characteristics of the Tactical Internet	Resulting Requirements for Multicast Routing Protocols
Limited Bandwidth	• minimize control traffic overhead • minimize traffic sent to "off tree" routers
Changing Topology	for protocols which maintain "soft" state - adapt quickly to changes in topology
Likely Disruption or Destruction of Links	• better distribute traffic over links • avoid central points of failure

Table 6.1. Requirements for Network Layer Multicast Routing Protocols Operating in the Tactical Internet.

No single protocol satisfied all the requirements levied by the tactical internet. Which protocols are chosen for TDN depends on how the decision maker weights the requirements imposed by the tactical internet.

The characteristics of applications hosted on the tactical internet will also influence the choice of a network layer multicast routing protocol. If large numbers of senders and receivers are expected, the protocols chosen must be scalable. Because some protocols induce more end-to-end delay than others, how tolerant applications are of this delay will further narrow the list of possible contenders.

2.   Reliable Multicast Protocols

Transport layer reliable multicast protocols are similarly constrained by the conditions of the tactical internet. Table 6.2 summarizes the requirements that must be met by a reliable multicast protocol which operates over the tactical internet.

Characteristics of the Tactical Internet or its Users	Resulting Requirements for Transport Layer Reliable Multicast Protocols
Limited Bandwidth	minimize protocol overhead - determined primarily by the method of error control
Likely Disruption or Destruction of Links	provide for failure recovery (if the application requires it)
High Bit Error Rates	assume that most errors are caused by the network rather than by receivers

Table 6.2. Requirements for Transport Layer Reliable Routing Protocols Operating in the Tactical Internet.

Since reliable multicast protocols are designed for end systems, the application will also heavily influence the selection process. Some applications may require a more relaxed form of reliability, while others may demand stronger guarantees. Applications may also differ in the level of ordering they require from a reliable multicast protocol. For some applications, failure recovery may be a responsibility of the protocol, or it may be handled by the application. The reliable multicast protocols which are selected must meet the demands of the application for which they were designed while still operating within the constraints imposed by the tactical internet.

B. RECOMMENDATIONS

1. Expand This Study

a. Include Other Reliable Multicast Protocols

To make this study more complete, several other reliable multicast protocols need to be evaluated. Candidate protocols include, but are not limited to: the Adaptive File Distribution Protocol (AFDP), ISIS and its successor HORUS, the Reliable Multicast Transport Protocol (RMTP) developed by NTT, Single Connection Emulation (SCE), Local Group Concept (LGC), and TOTEM.

b. Include Other Multicast Routing Protocols

In this study, only well known IETF multicast routing protocols were evaluated. Recent work in minimal cost spanning trees should be evaluated, and other emerging approaches to multicast routing should also be considered.

c.   *Focus on Implementations*

This study has focused primarily on the technical aspects of both network and transport layer multicast protocols. To better assess the commercial acceptance of these protocols, and the likelihood of their emergence as de facto industry standards, more details should be sought concerning the implementations of both types of protocols.

d.   *Study the Application Requirements of Tactical Data Systems*

Since applications influence the type of multicast protocol selected, tactical data systems should be studied more closely to determine the demands they may make of these protocols. Any such study should expand upon the work in this area begun by (Macker, 1996).

e.   *Include Other Network Technologies*

In the future, some portions of the tactical internet may be switched by Asynchronous Transfer Mode (ATM); commercial wireless protocols are sure to become prevalent as well. How these technologies multicast, and how their introduction will impact current multicasting protocols needs to be researched.

2.   Form Alliances With Protocol Developers

The Marine Corps recognizes that open solutions should be adopted, to both save money and promote interoperability. However, none of the reliable multicast protocols has yet emerged as a standard (in both the de jure and de facto senses). To ensure that its needs are represented as these next generation protocols are developed, the Marine Corps should adopt the position recommended by Buddenberg for the Navy:

> In general, what the Navy should be doing is defining its protocol requirements and encouraging working groups in public, vendor neutral fora to work out the standards and implementations. We should encourage the protocols and implementations in the public domain so that the Navy can buy them at COTS prices. (Buddenberg, 1996)

One way to make certain that Marine specific needs are represented is to implement and test the protocols that have been developed. The code for several of these protocols is publicly available; the code for other protocols is either proprietary or must be licensed. Feedback provided to developers will make them more familiar with Marine Corps requirements.

3. Monitor the Results of the Multicast Implementation STudy (MIST) Program

MIST is a program undertaken to investigate the multicast requirements of the Internet and DIS environments. The deliverable will be a "…common framework for reliable multicast…" that will be submitted to the IETF for approval. (MIST website) Monitoring the progress of this program will ensure that the Marine Corps is aware of those protocols which may become Internet standards.

4. Develop a Tactical Internet Testbed

Valid performance comparisons of reliable multicast protocols are impossible without a consistent test configuration. How each protocol will fare in a tactical internet is also difficult to assess if tests do not accurately represent the environment in which the protocols will be deployed. A testbed which closely resembles the tactical internet will allow meaningful comparisons to be made between protocols. How well suited a protocol is for the tactical internet will also be easier to determine from experiments conducted on such a testbed.

GLOSSARY

Although there may be several definitions for the terms in this glossary, only those related to a term's usage in this thesis are included here.

ACK - (ACKnowledgment). An ACK is the packet returned to the sender acknowledging receipt of its data.

Backbone - a common network to which other networks are connected. Although the term refers more to the role a network plays than to its capacity, backbone networks are generally of higher speed than those which connect to it. (Lynch, 1993)

Bandwidth - a measure of the capacity of a communications link. High bandwidth lines deliver more data in a unit time than lines with less bandwidth.

Broadcast - to transmit data to all possible receivers.

Buffer - the space in memory dedicated to storing network data.

Collaborative-Interactive Application - an application that facilitates people working together. Participants are usually both senders and receivers, and may communicate with each other through voice, video, or by sharing a common drawing space. Collaborative-Interactive Applications may also be called Distributed Collaborative Planning (DCP) applications.

Congestion - a condition caused by too many packets in a part of the network.

Connection-oriented - a term describing how data is transmitted from the sender to a receiver. In the connection-oriented communication model, a connection is established between the sender and receiver before data is transmitted. Data arrives at the receiver in the same order as it was sent, and the connection is released after all data has been transmitted.

Control Traffic - packets that contain information other than data. Control traffic has also been described as "bits about bits" by Negroponte (1995).

COTS - (Commercial Off The Shelf). Products developed commercially and available on the open market are often referred to as COTS products.

Datagram - a packet labeled with its destination address that is delivered without any guarantee of arriving at that destination.

Dense Environment - a network environment in which multicast group members are relatively densely packed and bandwidth is plentiful.

115

DIS - (Distributed Interactive Simulation). A simulation distributed across several computing platforms which involves participants in the simulation.

DISN - (Defense Information Systems Network). The DISN is a worldwide data network for DoD traffic.

Distributed Collaborative Planning Application - see Collaborative Interactive Application.

Distributed Processing Applications - applications in which multiple computers contribute to the completion of a task.

Domain - a region of the network.

End System - data producer or consumer attached to the network.

Error Control - the procedures to detect damaged, lost, or duplicate packets and recover from these errors.

Ethernet - a bus-based broadcast protocol for local area networks that operates at 10 Mbps. (Tanenbaum, 1996)

FDDI - (Fiber Distributed Data Interface). FDDI is a high performance, fiber based, token ring local area network protocol that operates at 100 Mbps. (Tanenbaum, 1996)

FEC - (Forward Error Correction). FEC is a method of error control that detects and corrects errors based upon redundant information included in the data stream. (Lucas, 1995)

FIFO - (First-In-First-Out). FIFO is a queuing strategy that services the first member of the queue first.

Flow Control - a method of restricting the amount of information that can be sent. (XTP Forum, 1995)

Go-Back-N Retransmission - a method of retransmission in which all data after a certain sequence number is resent, even if this data includes sequence numbers that were received correctly before.

Groupware - software that supports collaborative group activity.

Handshake - an exchange of messages between two hosts. The initial message is repeated until a response is elicited from the receiving host. (XTP Forum)

Hub - a common central device to which other hubs, hosts, or devices can be attached.

HMMWV - (High Mobility Multi-Wheeled Vehicle or "Hummer"). The HMMWV is the standard low tonnage tactical all-terrain vehicle for the Marine Corps.

Host - a computing device. See end system.

Interactive Collaborative Application - see Collaborative-Interactive Application.

Internet - a network of networks. Internet with a lower case "i" refers to a non-specific network of networks; internet with an upper case "i" is the world wide mesh that is frequently called the "information superhighway."

IP - (Internet Protocol). The Internet Protocol defines the format of the datagram and a global addressing scheme. IP is the network layer protocol for the commercial Internet.

ISDN - (Integrated Services Digital Network). A network that provides end-to-end digital connectivity; phone companies sell access to this network.

ISO - (International Organization for Standardization). ISO is an international level organization that creates standards for computer-to-computer communications.

ISO Stack - also called the Open Systems Interconnect (OSI) reference model. The OSI reference model is an abstraction designed to simplify the complexity of data communications, but was originally a way of organizing the different committees within ISO.

JTF - (Joint Task Force). A JTF is a force assembled from components of the different services.

LAN - (Local Area Network). A LAN is a network with a diameter of less than a few kilometers.

Latency - delay.

MAC - (Media Access Control). MAC protocols belong to the MAC sublayer of the data link layer of the ISO reference model and determine the order of transmission onto a shared medium (such as a LAN). (Tanenbaum, 1996)

MBone - (Multicast Backbone). The MBone is a virtual network of routers that support multicast.

Multicast Delivery Tree - the path established by a multicast routing protocol from a multicast source to members of the multicast group.

Multicast Group - a sender and its multicast receivers.

Multichannel Radio - a radio which is combined with a multiplexer to allow multiple sources to send over a single channel.

Multiplexing - either combining the data from multiple sources into one stream, or distributing the data from a single source onto multiple streams.

NACK - (Negative ACKnowledgment; also written as NAK). A packet returned to the sender indicating that data has been lost.

Network - computers which communicate with each other over a shared medium.

NIC - (Network Interface Card). The integrated circuit board which connects a host to the network.

Overhead - the cost, in bandwidth, of control traffic.

Packet-Switched Network - a wide area network composed of routers that store each packet until the required outgoing line becomes available.

Partition - that portion of a multicast group separated from the rest of the group by a network failure.

Point-to-Point Link - a link between two geographically separated communicants.

Protocol - common rules of communication.

Quality of Service - the guarantees that a protocol makes with regard to reliability, delay, and other communication parameters.

Rate Control - the method of limiting the rate at which a sender can transmit data. (XTP Forum, 1995)

Router - a network device that directs traffic to its destination.

RTT - (Round Trip Time). RTT is defined as the time beginning with the transmission of a packet and ending with the receipt of an acknowledgment for the same packet.

Scaling - how well a protocol adjusts to an increase in the number of senders or receivers.

Scope - the reach in a network. Limited scope implies that only a relatively small area of the network is affected, while something with global scope affects the entire network.

Selective Retransmission - a method of retransmission in which only damaged or lost packets are resent.

Sequence Number - a number assigned to a byte in the data stream which increases monotonically for consecutive bytes.

Session - a data communications conversation.

Single Channel Radio - a radio which transmits data from one source over its single channel.

Simplex - a mode of communication in which data travels in only one direction.

Sparse Environment - a network environment in which multicast group members are distributed and bandwidth is limited.

State - costs to routers, which include: (1) memory required for forwarding tables and configuration information, and (2) the costs associated with calculating this information. (Crowcroft, 1996) State may also be defined as the amount of memory which a host uses to store data associated with managing communications.

TCP - (Transmission Control Protocol). A transport layer protocol that guarantees reliability. TCP is frequently juxtaposed with IP (TCP/IP) to indicate that it guarantees reliability for IP datagrams.

Token - the right to access the transmission medium.

Token Bus - a protocol for local area networks that requires hosts to possess a token before transmitting. Although hosts for a Token Bus LAN are connected to a linear cable, the LAN is logically organized into a ring. (Tanenbaum, 1996)

Token Ring - a local area network protocol that circulates a special bit pattern, called a token, around a logical ring. To transmit, hosts must first acquire the token. (Tanenbaum, 1996)

TTL - (Time to Live). TTL is a method of limiting the scope of a packet.

Unicast - communication between a source and a single receiver.

WAN - (Wide Area Network). A network of geographically dispersed LAN's.

LIST OF REFERENCES

Aiello, Rosario, Pagani, Elena, and Rossi, Gian Paolo, *Causal Ordering in Reliable Group Communications*, Proceedings SIGCOMM '93 Conference, ACM, pp. 106 - 115, 1993.

Armstrong, S., Freier, A., and Marzullo, K., *Multicast Transport Protocol*, Request for Comments 1301, Internet Engineering Task Force (IETF), February 1992. Also available at *ftp://ds.internic.net/rfc/rfc1301.txt*

Atwood, J. William, et. al., *Reliable Multicasting in the Xpress Transport Protocol*, XTP Forum, 1996.

Ballardie, A. J., *Core Based Trees (CBT) Multicast Architecture*, Inter-Domain Multicast Routing (IDMR) Internet Draft, February 9, 1996. Available at *ftp://ftp.ietf.org/internet-drafts/draft-ietf-idmr-cbt-arch-03.txt*

Ballardie, Tony, Francis, Paul, and Crowcraft, Jon, *Core Based Trees (CBT): An Architecture for Scalable Inter-Domain Multicast Routing*, Proceedings SIGCOMM '93 Conference, ACM, pp. 85 - 95, 1993.

Bormann, C., et. al., *MTP-2: Towards Achieving the S.E.R.O. Properties for Multicast Transport.* Presented at the ICCCN '94, San Francisco, September 1994. Also available at *http://hill.lut.ac.uk/DS-Archive/MTP.html*

Braudes, R., and Zabele, S., *Requirements for Multicast Protocols*, Request for Comments 1458, Internet Engineering Task Force (IETF), May 1993. Also available at *ftp://ds.internic.net/rfc/rfc1458.txt*

Buddenberg, Rex A., electronic mail message to Don McGregor, July 31, 1996.

Buddenberg, Rex A., et. al., *Internetwork Developments and Their Impacts on Command and Control Systems*, Proceedings of the 1996 Command and Control Research and Technology Symposium, Naval Postgraduate School, Monterey, California, Jun 25 - 28, 1996, pp. 553 - 562.

Callahan, Jack, Montgomery, Todd, and Whetten, Brian, *High-Performance, Reliable Multicasting: Foundations for Future Internet Groupware Applications*, 1995. Available at *http://research.ivv.nasa.gov/~callahan/Papers/net2001l/net2001l.html*

Clark, David D., and Tennenhouse, David L., *Architectural Considerations for a New Generation of Protocols*, Proceedings of ACM SIGCOMM, September 1990, pp. 201 - 208.

Crowcroft, Jon, Wakeman, Ian, and Handley, Mark, *Internetworking Multimedia - Going With the Flow,* Work in Progress, 1996. Available at *http://www.cs.ucl.ac.uk/staff/jon/mmbook/book/book.html*

Dismuke, Jerry B., *Formal Specification, Verification, and Analysis of the Reliable Multicast Transport Protocol*, Master's Thesis, Naval Postgraduate School, December 1995.

Deering, Stephen, et.al., *Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification*, June 6, 1996. Available at *ftp://ds.internic.net/internet-drafts/draft-ietf-idmr-pim-sm-spec-05.txt*

Deichler, E. C., and Fellows, B. D., *Modeling The Communications Architecture for the Marine Air-Ground Task Force*, Proceedings of the 1996 Command and Control Research and Technology Symposium, Naval Postgraduate School, Monterey, California, Jun 25 - 28, 1996, pp. 635 - 661.

Dempsey, Bert J., and Alfred C. Weaver, *Issues in Designing Transport Layer Multicast Facilities*, Computer Science Report No. TR-90-18, University of Virginia, July 16, 1990. Also available at *http://hill.lut.ac.uk/DS-Archive/MTP.html*

Estrin, Deborah, *Lecture notes for CSCI 551, Computer Communications*, USC, 1996. Available at *http://www-scf.usc.edu/~dbyrne/*

Estrin, D., et. al., *Protocol Independent Multicast (PIM), Dense Mode Protocol Specification*, Jan 17, 1996. Available at *ftp://ds.internic.net/internet-drafts/draft-ietf-idmr-pim-dm-spec-03.txt*

Floyd, Sally, et. al., *A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing*, ACM SIGCOMM 95, pp. 342-356, August 1995. Also available at *ftp://ftp.ee.lbl.gov/papers/link.ps.Z*

Garcia-Molina, Hector, and Spauster, Annemarie, *Ordered and Reliable Multicast Communication*, ACM Transactions on Computer Systems, Vol. 9, No. 3, August 1991, pp. 242 - 272.

Heybey, Andrew Tyrrell, *Video Coding and the Application Level Framing Protocol Architecture*, Master's Thesis, Massachusetts Institute of Technology, June, 1991. Also available at *http://ana-www.lcs.mit.edu/anaweb/abstracts/tr-542.html*

Kaashoek, Frans M., and Tanenbaum, Andrew S., *Efficient Reliable Group Communication for Distributed Systems*, Report IR-295, Vrije Universiteit, July, 1992. Also available at *http://www.cs.vu.nl/fb/generated/publicaties/Tanenbaum.jaarv92.html*

122

Koifman, A., and Zabele, S., *RAMP: A Reliable Adaptive Multicast Protocol*. Submitted to INFOCOM '96, San Francisco, CA, Mar, 1996. Also available at *http://www.tasc.com:80/simweb/papers/RAMP/abstract.htm*

Lin, John C., and Paul, Sanjoy, *RMTP: A Reliable Multicast Transport Protocol*, Proceedings of IEEE INFOCOM '96, March 1996, pp. 1414 - 1424. Also available at *http://hill.lut.ac.uk/DS-Archive/MTP.html*

Lucas, Matthew T., Dempsey, Bert J., and Weaver, Alfred C., *Distributed Error Recovery for Continuous Media Data in Wide-Area Multicast*, University of Virginia Technical Report CS95-52, July 18, 1995. Also available at *http://hill.lut.ac.uk/DS-Archive/MTP.html*

Lynch, Daniel C., and Rose, Marshall T., *Internet System Handbook*, Reading, Massachusetts, Addison-Wesley Publishing Company, Inc., 1993.

Macker, Joseph P., Klinker, J. Eric, and Corson, M. Scott, *Reliable Multicast Data Delivery for Military Networking*, March 1996. Submitted to IEEE MILCOM '96 Conference.

Macker, Joseph P., electronic mail message to the author, May 22, 1996.

Macker, Joseph P., electronic mail message to the author, August 8, 1996.

Marine Corps Combat Development Command (MCCDC), *United States Marine Corps Technical Architecture version 1.0*, Quantico, Virginia, 5 October 1995.

Marine Corps Systems Command (MARCORSYSCOM), *Tactical Data Network System Description*, Quantico, Virginia, September 1995.

Mayer, Erwin, *An Evaluation Framework for Multicast Ordering Protocols*, Proceedings ACM SIGCOMM '92, Baltimore, Maryland, pp. 177 - 187.

Multicast Implementation STudy (MIST) Program, *Reliable Multicast Protocol Web Site*, *http://www.tascnets.com/mist/doc/mcpCompare.html*

Montgomery, Todd, *Design, Implementation, and Verification of the Reliable Multicast Protocol*, Master's Thesis, West Virginia University, 1994.

Montgomery, Todd, Whetten, Brian, and Callahan, John R., *The Reliable Multicast Protocol Specification Flow Control and NACK Policy*, 5 October, 1995. Also available at *ftp://research.ivv.nasa.gov/pub/doc/RMP/RMPflow.txt*

Montgomery, Todd, electronic mail message to Jon Crowcroft, March 8, 1996. Also available at *http://www.pearnet.org/hypernews/EColabor/ml/199606/33.html*

Montgomery, Todd, electronic mail message to the author, August 21, 1996.

Negroponte, Nicholas, *being digital*, New York, New York, Alfred A Knopf, Inc., 1995.

Nierle, James E., *Internetworking: Technical Strategy for Implementing the Next Generation Internet Protocol (IPv6) in the Marine Corps Tactical Data Network*, Master's Thesis, Naval Postgraduate School, June, 1996. Also available at *http://www.stl.nps.navy.mil/~jenierle/thesis.html*

Paul, Sanjoy, electronic mail message to the author, June 24, 1996.

Pingali, Sridhar, Towsley, Don, and Kurose, James F., *A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols*, Performance Evaluation Review, Vol. 22, No. 1, May 1, 1994, pp. 221 - 230.

Rajagopalan, Bala, *Reliability and Scaling Issues in Multicast Communications*, Proceedings SIGCOMM '92 Conference, ACM, pp. 188 - 198, 1992.

Saltzer, J. H., Reed, D. P., and Clark, D. D., *End-To-End Arguments in System Design*, ACM Transactions on Computer Systems, Vol. 2, No. 4, November 1984, pp. 277 - 288.

Semeria, C., and Maufer, T., *Introduction to Multicast Routing*, Internet Draft, Internet Engineering Task Force (IETF), March 1996. Available at *ftp://ds.internic.net/internet-drafts/draft-rfced-info-semeria-00.txt*

Sudan, Madhu, electronic mail message to the author, May 21, 1996.

Tanenbaum, Andrew S., *Computer Networks, 3rd ed.*, Upper Saddle River, New Jersey, Prentice Hall PTR, 1996.

Voigt, Robert J., *A Hierarchical Approach to Multicast in a Datagram Internetwork*, PhD Dissertation, Naval Postgraduate School, March 1996.

Webcast website, http://www.ncsa.uiuc.edu/SDG/Software/XMosaic/CCI/webcast-new.html

Wei, Liming, and Estrin, Deborah, *The Trade-offs of Multicast Trees and Algorithms*, Proceedings of the 1994 International Conference on Computer Communications and Networks. Available at *http://www.usc.edu/dept/cs/tech.html*

Whetten, Brian, Montgomery, Todd, and Kaplan, Simon, *A High Performance Totally Ordered Multicast Protocol*, 1995. Available at *http://hill.lut.ac.uk/DS-Archive/MTP.html*

XTP Forum, *Xpress Transport Protocol Specification, XTP Revision 4.0*, XTP Forum Inc., Santa Barbara, CA, March 1, 1995.

Xu, Ya. *Solutions to CSCI 551 Homework*, USC, 1996. Available at *http://netweb.usc.edu/yaxu/551/tony/*

Yavatkar, Rajendra, Griffioen, James, and Sudan, Madhu, *A Reliable Dissemination Protocol for Interactive Collaborative Applications*, 1995. Available at *http://hill.lut.ac.uk/DS-Archive/MTP.html*

126

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
    8725 John J. Kingman Rd., STE 0944
    Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
    Naval Postgraduate School
    411 Dyer Rd.
    Monterey, California 93943-5101

3. Director, Marine Corps Research Center 2
    MCCDC, Code C40RC
    2040 Broadway Street
    Quantico, VA 22134-5107

4. Director, Studies and Analysis Division 1
    MCCDC, Code C45
    3300 Russell Road
    Quantico, VA 22134-5130

5. Director, Training and Education 1
    MCCDC, Code C46
    1019 Elliot Rd.
    Quantico, VA 22134-5027

6. Professor Rex A. Buddenberg 1
    Code SM/Bu
    Naval Postgraduate School
    Monterey, California 93943-5101

7. Professor Suresh Sridhar 1
    Code SM/Sr
    Naval Postgraduate School
    Monterey, California 93943-5101

8. Professor Barry Frew 1
    Code SM/Fr
    Naval Postgraduate School
    Monterey, California 93943-5101